PETIR
JURNAL
PENGKAJIAN DAN PENERAPAN
TEKNIK INFORMATIKA

VOLUME 8 - NOMOR 1 | MARET 2015 | ISSN 1978-9262

ISSN 1978-9262
771978 926272

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)					
PETIR	VOL. 8	NO. 1	HAL. 1 - 132	JAKARTA, MARET 2015	ISSN 1978-9262

DETEKSI MATA REAL TIME MENGGUNAKAN OPENCV UNTUK ANDROID

Yustika Erliani[1], Bagus Priambodo[2],
[1,2] *Program Studi Sistem Informasi Universitas Mercu Buana*
Kampus Meruya Jakarta Barat
yustika.erliani@mercubuana.ac.id, bagus.priambodo@mercubuana.ac.id

Abstrak

Saat ini telah banyak berkembang aplikasi-aplikasi yang menggunakan fitur deteksi wajah. Deteksi wajah sendiri dapat dilakukan dengan berbagai cara, salah satunya menggunakan metode Viola-Jones, yaitu metode yang menggabungkan support vector machines, algoritma boosting, dan cascade classifie. Pada penelitian ini dirancang suatu sistem deteksi mata dengan menggunakan metode Viola-Jones dengan library OpenCV. Data dari penelitian ini berupa video live yang di ambil menggunakan camera smartphone dengan os android

***Kata Kunci** : deteksi mata, opencv, android.*

Abstract

Right now there has been a lot of developing applications that use the face detection feature. Face detection itself can be done in various ways, one using the Viola-Jones method, a method that combines support vector machines, boosting algorithm, and a cascade classifie. In this research designed an eye detection system by using a Viola-Jones method with OpenCV library. Data from this research are in the form of a live video capture using camera smartphone with android os

***Keyword:** eye detection, OpenCV, android.*

I. PENDAHULUAN

Saat ini telah banyak berkembang aplikasi-aplikasi yang menggunakan fitur deteksi wajah. Deteksi wajah sendiri dapat dilakukan dengan berbagai cara, salah satunya menggunakan metode *Viola-Jones*, yaitu metode yang menggabungkan *support vector machines*, algoritma *boosting*, dan *cascade classifier*. Kemudian metode ini diterapkan pada suatu sembarang citra digital, untuk mendapatkan posisi-posisi wajah pada citra tersebut. Metode ini relative mendapatkan hasil yang cepat, akurat, dan efisien. Metode *Viola-J*ones merupakan algoritma yang paling banyak digunakan untuk mendeteksi wajah

Proses pendeteksian wajah dilakukan dengan mengklasifikasikan sebuah gambar setelah sebelumnya sebuah pengklasifikasi dibentuk dari data latih.

OpenCV (Open Source Computer Vision Library: http://opencv.org) adalah BSD-berlisensi perpustakaan open source yang mencakup beberapa ratusan algoritma visi komputer. Dokumen ini menjelaskan apa yang disebut OpenCV 2.x API, yang pada dasarnya adalah C ++ API, sebagai berlawanan dengan berbasis C OpenCV 1.x API.

Pada penelitian ini dirancang suatu sistem deteksi mata dengan menggunakan metode *Viola-Jones* dengan library OpenCV. Data dari penelitian ini berupa video live yang di ambil menggunakan camera smartphone dengan os android.

II. RUMUSAN MASALAH

Perumusan masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana membuat aplikasi yang dapat mendeteksi mata dari camera smartphone secara real time?
2. Bagaimana opencv diimplementasikan untuk mendeteksi mata secara real time pada smartphone android?

III. BATASAN MASALAH

Masalah yang akan dibahas pada penelitian kali hanya menampilkan video live

camera dan memberikan kotak berwarna hijau pada mata yang terdeteksi, tidak menyimpan gambar hasil deteksi ke dalam media storage

IV. TUJUAN

Adapun tujuan dari pembuatan penelitian ini adalah:

1. membuat aplikasi deteksi mata dari camera smartphone secara real time.
2. Mengetahui apakah library deteksi wajah opencv dapat digunakan dengan baik untuk deteksi mata
3. Mengetahui apakah library pengenalan wajah opencv dapat dikembangkan untuk deteksi senyum, deteksi marah dan pengenalan wajah lainnya

V. MANFAAT

Dengan membuat penelitian deteksi mata, dapat dilanjutkan dengan penelitian mendeteksi senyum, deteksi marah dan lain sebagainya

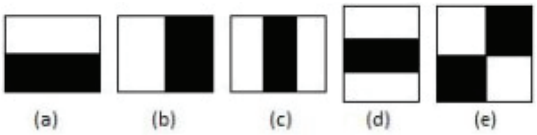
VI. LANDASAN TEORITIS

A. Algoritma Deteksi Wajah

Saat ini telah banyak berkembang aplikasi-aplikasi yang menggunakan fitur deteksi wajah. Deteksi wajah sendiri dapat dilakukan dengan berbagai cara, salah satunya menggunakan algoritma Viola-Jones.

Algoritma Viola-Jones (Viola dkk, 2001) (Viola dkk, 2004) merupakan algoritma yang paling banyak digunakan untuk mendeteksi wajah. Proses pendeteksian wajah dilakukan dengan mengklasifikasikan sebuah gambar setelah sebelumnya sebuah pengklasifikasi dibentuk dari data latih. Data latih yang digunakan oleh algoritma ini berjumlah 5000 citra wajah dan 9400 citra non-wajah sehingga menghasilkan akurasi sistem sebesar 95% dengan data positif salah sebesar 1 : 14084.

Klasifikasi citra dilakukan berdasarkan nilai dari sebuah fitur. Penggunaan fitur dilakukan karena pemrosesan fitur berlangsung lebih cepat dibandingkan pemrosesan citra perpiksel. Terdapat 3 jenis fitur berdasarkan jumlah persegi panjang yang terdapat di dalamnya, seperti yang dapat dilihat pada gambar di bawah ini :



Gambar 2. Gambar tepi dan line

Pada gambar di atas dapat dilihat bahwa fitur (a) dan (b) terdiri dari dua persegi panjang, sedangkan fitur (c) dan (d) terdiri dari tiga persegi panjang dan fitur (e) empat persegi panjang. Cara menghitung nilai dari fitur ini adalah mengurangkan nilai piksel pada area hitam dengan piksel pada area putih. Untuk mempermudah proses penghitungan nilai fitur, algoritma Viola-Jones menggunakan sebuah media berupa citra integral.

Citra integral adalah sebuah citra yang nilai tiap pikselnya merupakan akumulasi dari nilai piksel atas dan kirinya. Sebagai contoh, piksel (a,b) memiliki nilai akumulatif untuk semua piksel (x,y) dimana $x \leq a$ dan $y \leq b$. Contoh citra integral dapat dilihat di bawah :

1	2	3
4	5	6
7	8	9

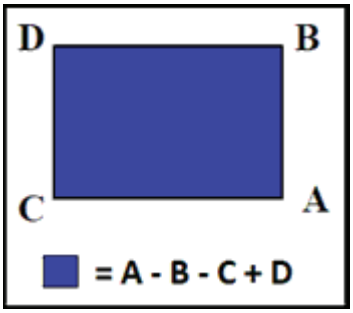
Citra Masukan

1	3	6
5	12	21
12	27	45

Citra Integral

Gambar 3. Citra masukan dan citra integral

Dengan mendapatkan citra integral, penghitungan nilai area dapat dilakukan dengan cara :



Gambar 4. Penghitungan nilai Area

Proses pencarian nilai fitur ini dilakukan secara iteratif mulai dari ujung kiri atas citra hingga ujung kanan bawah dengan pergeseran sebesar Δx dan Δy. Semakin kecil nilai Δx dan Δy, maka semakin akurat pula proses deteksi. Nilai Δx dan Δy yang sering digunakan adalah 1.

Penghitungan nilai fitur diulangi hingga 12 kali dengan skala 1,25. Pemrosesan fitur terbaik diperoleh pada subcitra berukuran 24 x 24 sehingga pada setiap iterasi, subcitra

dikonversi terlebih dahulu menjadi berkuran 24 x 24. Sebagai contoh, pertama-tama iterasi dilakukan dengan ukuran subcitra 24 x 24. Subcitra ini tidak perlu dikonversi sehingga langsung dapat dilakukan penghitungan nilai fitur. Setelah iterasi tahap ini selesai, dilakukan iterasi dengan subcitra berukuran 1,25 x 24 = 30. Subcitra berukuran 30 x 30 dikonversi menjadi 24 x 24 sebelum dilakukan penghitungan nilai fitur. Proses ini diulangi hingga 12 kali.

Permasalahan yang terdapat dalam penghitungan fitur ini adalah Viola – Jones memiliki 60.000 jenis fitur yang berbeda. Jumlah ini terlalu besar sehingga tidak mungkin dilakukan penghitungan untuk semua fitur. Hanya fitur-fitur tertentu sajalah yang dipilih untuk diikutsertakan. Pemilihan fitur-fitur ini dilakukan menggunakan algoritma Ada-Boost.

Algoritma Ada-Boost berfungsi untuk mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap data latih dengan menggunakan nilai dari fitur tersebut. Fitur yang memiliki batas terbesar antara wajah dan non-wajah dianggap sebagai fitur terbaik.

Karakteristik dari algoritma Viola-Jones adalah adanya klasifikasi bertingkat. Klasifikasi pada algoritmna ini terdiri dari 3 tingkatan dimana tiap tingkatan mengeluarkan subcitra yang diyakini bukan wajah. Hal ini dilakukan karena lebih mudah untuk menilai subcitra tersebut bukan wajah ketimbang menilai apakah subcitra tersebut berisi wajah. Di bawah ini adalah alur kerja dari klasifikasi bertingkat.

Pada klasifikasi tingkat pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Klasifikasi ini kira-kira akan menyisakan 50% subcitra untuk diklasifikasi di tahap kedua. Seiring dengan bertambahnya tingkatan klasifikasi, maka diperlukan syarat yang lebih spesifik sehingga fitur yang digunakan menjadi lebih banyak. Jumlah subcitra yang lolos klasifikasi pun akan berkurang hingga mencapai jumlah sekitar 2%..

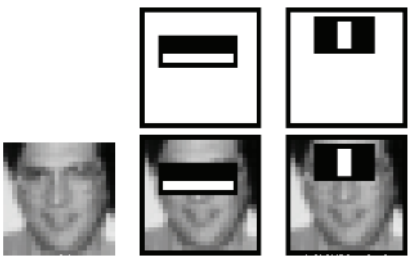
B. Deteksi Objek menggunakan Haar

Deteksi Objek menggunakan Haar fitur berbasis pengklasifikasi cascade merupakan metode deteksi obyek efektif diusulkan oleh Paul Viola dan Michael Jones dalam makalah mereka, "Object Detection cepat menggunakan Cascade Didorong Fitur Simple" pada tahun 2001 Ini adalah pendekatan berbasis pembelajaran mesin di mana fungsi cascade dilatih dari banyak gambar positif dan negatif. Hal ini kemudian digunakan untuk mendeteksi benda-benda di gambar lainnya.

Di sini kita akan bekerja dengan deteksi wajah. Awalnya, algoritma membutuhkan banyak gambar positif (gambar wajah) dan citra negatif (gambar tanpa wajah) untuk melatih classifier. Kemudian kita perlu untuk mengambil fitur dari itu. Untuk ini, fitur haar yang ditunjukkan pada gambar di bawah ini digunakan. Mereka seperti kernel convolutional kami. Setiap fitur adalah nilai tunggal yang diperoleh dengan mengurangi jumlah piksel di bawah persegi panjang putih dari jumlah piksel di bawah persegi panjang hitam.

Sekarang semua kemungkinan ukuran dan lokasi masing-masing kernel digunakan untuk menghitung banyak fitur. (Coba bayangkan berapa banyak perhitungan yang dibutuhkan? Bahkan hasil jendela 24x24 lebih dari 160.000 fitur). Untuk setiap perhitungan fitur, kita perlu menemukan jumlah piksel di bawah persegi putih dan hitam. Untuk mengatasi ini, mereka memperkenalkan gambar terpisahkan. Ini menyederhanakan perhitungan jumlah piksel, seberapa besar mungkin jumlah piksel, untuk sebuah operasi yang melibatkan hanya empat piksel. Bagus, bukan? Itu membuat hal-hal yang super cepat.

Tapi di antara semua fitur ini kita menghitung, sebagian besar dari mereka tidak relevan. Sebagai contoh, perhatikan gambar di bawah ini. Top baris menunjukkan dua fitur yang baik. Fitur pertama yang dipilih tampaknya fokus pada properti yang wilayah mata sering lebih gelap dari daerah hidung dan pipi. Fitur kedua yang dipilih bergantung pada properti yang mata lebih gelap dari jembatan hidung. Tapi jendela yang sama berlaku pada pipi atau tempat lain tidak relevan. Jadi bagaimana kita memilih fitur terbaik dari 160000+ fitur? Hal ini dicapai dengan AdaBoost.



Gambar 5. Wajah dan deteksi

Untuk ini, kita menerapkan setiap fitur pada semua gambar pelatihan. Untuk setiap fitur, ia menemukan ambang terbaik yang akan mengklasifikasikan wajah untuk positif dan negatif. Tapi jelas, akan ada kesalahan atau misclassifications. Kami memilih fitur dengan tingkat kesalahan minimum, yang berarti mereka adalah fitur yang paling mengklasifikasikan wajah dan non-wajah gambar. (Proses ini tidak sesederhana ini. Setiap gambar diberi bobot yang sama pada awal. Setelah setiap klasifikasi, bobot gambar terkelompokan meningkat. Lalu proses lagi sama dilakukan. Tingkat kesalahan baru dihitung. Juga bobot baru. Proses dilanjutkan sampai diperlukan akurasi atau kesalahan tingkat dicapai atau diperlukan sejumlah fitur yang ditemukan).

Classifier Akhir adalah jumlah tertimbang ini pengklasifikasi lemah. Hal ini disebut lemah karena saja tidak dapat mengklasifikasikan gambar, tetapi bersama-sama dengan yang lain membentuk classifier yang kuat. Jurnal mengatakan bahkan 200 fitur menyediakan deteksi dengan akurasi 95%. Pengaturan terakhir mereka memiliki sekitar 6000 fitur. (Bayangkan pengurangan dari 160000+ fitur untuk 6000 fitur. Itu adalah keuntungan besar).

C. *OpenCV*

OpenCV (Open Source Computer Vision Library: http://opencv.org) adalah BSD-berlisensi perpustakaan open source yang mencakup beberapa ratusan algoritma visi komputer. Dokumen ini menjelaskan apa yang disebut OpenCV 2.x API, yang pada dasarnya adalah C ++ API, sebagai berlawanan dengan berbasis C OpenCV 1.x API. Yang terakhir ini dijelaskan dalam opencv1x.pdf.

OpenCV memiliki struktur modular, yang berarti bahwa paket tersebut termasuk beberapa perpustakaan bersama atau statis. Modul yang tersedia:

inti - modul kompak mendefinisikan struktur data dasar, termasuk fungsi padat multi-dimensi array Mat dan dasar yang digunakan oleh semua modul lainnya.

imgproc - modul pengolahan citra yang mencakup linear dan gambar non-linear filtering, transformasi gambar geometris (mengubah ukuran, affine dan perspektif warping, berbasis tabel generik remapping), konversi ruang warna, histogram, dan sebagainya.

Video - modul analisis video yang mencakup estimasi gerak, latar belakang pengurangan, dan algoritma pelacakan objek.

calib3d - dasar multi-lihat algoritma geometri, kamera kalibrasi tunggal dan stereo, obyek menimbulkan estimasi, algoritma stereo korespondensi, dan unsur-unsur rekonstruksi 3D.

features2d - detektor yang menonjol, deskripsi, dan pencocokan deskripsi.

objdetect - deteksi objek dan contoh dari kelas yang sudah ditetapkan (misalnya, wajah, mata, mug, orang, mobil, dan sebagainya).

highgui - mudah digunakan antarmuka untuk merekam video, gambar dan video codec, serta kemampuan UI sederhana.

gpu - algoritma GPU-accelerated dari modul OpenCV yang berbeda.

... Beberapa modul pembantu lainnya, seperti Flann dan Google pembungkus tes, Python binding, dan lain-lain

VII. *ANALISA DAN PEMBAHASAN*

Metode penelitian yang akan dilakukan adalah sebagai berikut :

1. Studi literatur yang berhubungan dengan penelitian
2. Menyiapkan software yang akan digunakan untuk penelitian.
3. Menyiapkan data input dan algoritma untuk deteksi mata.
4. Menyiapkan data training / classifier dan detection.
5. Deteksi mata dari video real time.
6. Gambar kotak pada video pada mata yang terdeteksi.
7. Membuat kesimpulan dan saran.

Menyiapkan perangkat lunak

Perangkat lunak yang digunakan pada penelitian ini antara lain

1. OpenCV untuk android.
2. Eclipse Indigo.
3. SDK Android.

Menyiapkan data input dan algoritma deteksi mata

Data input yang akan digunakan adalah video live yang berasal dari camera smartphone.

Algoritma yang akan digunakan adalah Haar feature-based cascade classifiers

Data input berupa video, kemudian menggunakan machine learning ada boost

menggunakan data training haarclasifier akan dicari objek mata. Setelah ketemu akan didapat kotak mata. Kotak mata itu akan di gambar pada video live pada smartphone

Deteksi mata dari video real time

Proses deteksi mata ditunjukan pada method onCameraFrame pada source code berikut ini :

```
    public Mat
onCameraFrame(CvCameraViewFrame
inputFrame) {

        mRgba = inputFrame.rgba();
        mGray = inputFrame.gray();

        if (mAbsoluteFaceSize == 0) {
           int height = mGray.rows();
           if (Math.round(height *
mRelativeFaceSize) > 0) {
              mAbsoluteFaceSize =
Math.round(height * mRelativeFaceSize);
           }

mNativeDetector.setMinFaceSize(mAbsoluteFa
ceSize);
        }

        MatOfRect faces = new MatOfRect();

        if (mDetectorType ==
JAVA_DETECTOR) {
           if (mJavaDetector != null)

mJavaDetector.detectMultiScale(mGray, faces,
1.1, 2, 2,
                    new Size(mAbsoluteFaceSize,
mAbsoluteFaceSize), new Size());
        }
        else if (mDetectorType ==
NATIVE_DETECTOR) {
           if (mNativeDetector != null)
              mNativeDetector.detect(mGray,
faces);
        }
        else {

        }

  // Apabila array kotak mata lebih dari 1
atau ketemu maka cetak kotak

        Rect[] facesArray = faces.toArray();
        for (int i = 0; i < facesArray.length;
i++)
            Core.rectangle(mRgba,
facesArray[i].tl(), facesArray[i].br(),
FACE_RECT_COLOR, 3);

        return mRgba;
    }
```

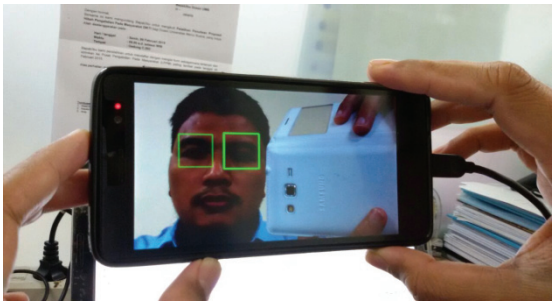
Training dengan Haar

```
 //InputStream is =
getResources().openRawResource(R.raw.lbpca
scade_frontalface);

InputStream is =
getResources().openRawResource(R.raw.haarc
ascade_lefteye_2splits);

File cascadeDir = getDir("cascade",
Context.MODE_PRIVATE);

// Cascade menggunakan frontalface

//mCascadeFile = new File(cascadeDir,
"lbpcascade_frontalface.xml");

// Diganti menggunakan
haarcascade_frontalface_alt

mCascadeFile = new File(cascadeDir,
"haarcascade_lefteye_2splits.xml");
```

Source code diatas menunjukan, aplikasi dijalankan menggunakan training haarcascade training. Kita dapat membuat data training sendiri untuk mendeteksi objek yang kita kehendaki seperti pesawat, mobil dan lain lain.

VIII. *IMPLEMENTASI*

Aplikasi di jalankan dan diuji apakah telah berjalan sesuai yang diinginkan



Gambar 6. Gambar Capture Hasil deteksi mata

Gambar. 6 menujukkan hasil capture dari deteksi mata pada aplikasi yang berjalan pada os Android

IX. KESIMPULAN DAN SARAN

Kesimpulan

Dari hasil uji yang dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi dapat digunakan untuk deteksi mata.
2. Library OpenCV deteksi mata dapat digunakan dengan baik.
3. Dapat diketahui dari hasil analisa library openCV dan source code nya. Kita dapat membuat data training sendiri untuk deteksi objek yang dikehendaki, sehingga memungkinkan membuat deteksi senyum atau deteksi marah menggunakan library openCV

Saran

Guna perbaikan dan memberikan masukan bagi penelitian selanjutnya, maka dapat disampaikan saran-saran ataupun rekomendasi sebagai berikut:

1. Penambahan aplikasi tidak hanya dapat untuk deteksi mata akan tetapi dapat digunakan untuk deteksi mulut.
2. Penelitian selanjutnya dapat ditambahkan data training untuk deteksi mimik wajah untuk mengenali mimik wajah. Misal nya gembira, dan sedih.

REFERENSI

[1] Viola, P., Jones, M. J., "Rapid Object Detection Using A Boosted Cascade of Simple Features", IEEE Conference on Computer Vision and Pattern Recognition, Jauai, Hawaii, 2001.

[2] Viola, P., Jones, M. J., "Robust Real-Time Face Detection", International Journal of Computer Vision, Kluwer Academic, Netherlands, 2004

[3] Dwisnanto Putro, Teguh Bharata Adji, Bondhan Winduratna, "Sistem Deteksi Wajah dengan Menggunakan Metode *Viola-Jones. Scietec 2012*

[4] OpenCV documentation, (http://opencv.org)