BANDUNG ANGKOT-FINDER

PERANCANGAN DAN IMPLEMENTASI MARKERLESS AUGMENTED REALITY PADA APLIKASI ANGKOT-FINDER DI KOTA BANDUNG UNTUK SMARTPHONE BERBASIS ANDROID

Deddy Gunawan [1], Astri Novianty [2], Surya Michrandi Nasution [3]
[123] Prodi S1 Sistem Komputer, Universitas Telkom Bandung
[1] deddygnwn14@gmail.com, [2] astri_nov@yahoo.com, [3] surya.michrandi@gmail.com

Abstrak—**Angkot (angkutan kota) adalah nama yang diberikan pada mobil yang berfungsi sebagai alat transportasi umum. Biasanya angkutan kota berukuran lebih kecil dari pada bus. Pada tulisan ini, angkutan kota digunakan sebagai pengganti marker untuk *augmented reality*. Idenya adalah untuk menjadikan objek sehari-hari seperti angkutan kota sebagai marker untuk augmented reality dan kemudian menginformasikan kepada pengguna rute apa yang diambil oleh angkutan kota yang tertangkap kamera. Hal ini membutuhkan deteksi objek, pengenalan objek, dan ekstraksi ciri. Haar *cascade* digunakan sebagai metoda untuk mendeteksi objek, Jaringan syaraf tiruan *Back Propagation* sebagai metoda untuk pengenalan objek, dan *Principal Component Analysis* sebagai metoda ekstraksi ciri. Sedangkan informasi objek dimunculkan sebagai *augmented reality*. Metoda *object recognition* yang diimplementasikan memiliki performansi yang tidak baik. Tingkat ketepatan informasi yang dimunculkan oleh augmented reality hanya sebatas 40% saja sehingga output sering berubah-ubah. Beberapa angkutan kota dengan warna tertentu justru mampu dikenali dengan akurasi yang cukup tinggi, namu beberapa angkutan kota lainnya bahkan tidak dapat dikenali sama sekali.**

Keywords—angkot; augmented reality;

Abstract—**Angkot (angkutan kota) or city transportation in english, is name given to a car that serve as a general transportation. Usually city transports are smaller than bus. In this paper, city transports are used as substitute of augmented reality marker. The idea is to make everyday object like city transport as a marker for augmented reality and then to inform the user what route is the city transport in the camera takes. This require object detection, object recognition, and feature extraction. Haar cascade of haar-like feature is used as object detection method, Back propagation Neural network as the object recognition method, and Principal Component Analysis to do feature extraction. The information about targeted object will be shown as augmented reality object. Object recognition method that implemented doesn't have good performance. The information that shown by augmented reality only 40% accurate so the output will change often. Some city transport with specific color sometimes can be recognized easily with high accuracy, but some other city transport are not even recognized at all.**

Keywords—angkot; augmented reality;

1. PENDAHULUAN

Bandung Angkot-Finder adalah aplikasi *smartphone* Android yang dibuat untuk tugas akhir ini. Tujuan dari aplikasi tersebut adalah untuk mempernudah pengguna mencari angkutan kota yang tepat sesuai harga dan rute tercepatnya. Aplikasi ini mampu mengidentifikasi angkutan kota berdasarkan bentuk dan warnanya denga menggunakan fitur *augmented reality*.

Angkutan kota di kota Bandung memiliki label di depan dan belakang, tapi tidak ada ketrangan mengenai rute apa yang angkutan kota tersebut lewati. Mungkin hal ini tidak masalah bagi orang yang sudah lama tinggal di bandung, tapi tidak bagi pendatang baru atau turis asing. Untuk tujuan itulah Bandung Angkot-Finder memiliki fitur untuk mengidentifikasi setiap angkutan kota di kota Bandung dengan mengrahkan kamera *smartphone* Android ke mobil angkutan kota dan informasi akan muncul sebagai objek *augmented reality*.

Keuntungan dari penelitian ini adalah angkutan kota di kota Bandung sudah memiliki keunikan pada cat di bodinya. Corak dari cat bisa dianggap sebagai ciri dari setiap jurusan angkutan kota, artinya ciri tersebut bisa diekstrak dengan menggunakan perhitungan *Prinicpal Component Analysis*. Dengan begitu tidak perlu lagi meletakkan marker pada setiap angkutan kota untuk tujuan *augmented reality*. Sehingga memungkinkan untuk mengambil keuntungan tersebut untuk membuat *markerless augmented reality*.

2. PEKERJAAN SEBELUMNYA

Ada beberapa teknik untuk membuat sebuah *augmented reality* tanpa marker. Seperti yang sudah di jelaskan pada –*Implementation of Augmented Reality System for Smartphone Advertisements*‖, sebuah jurnal yang yang ditulis oleh Young-geun Kim dan Won-jung Kim (2014) –according to the object-tracking techniques, augmented system is divided into positioning, marker, and markerless augmented reality services.‖ *Positioning augmented reality* sistemnya menggunakan posisi dari objek sebagai marker dan menggunakan GPS untuk melacak lokasinya. Pada paper tersebut dijelaskan kelemahan dari sistem tesebut. Ketika dua objek terdeteksi dekat satu sama lain, tampilannya akan berhimpitan dan menghalangi pandangan pengguna. Pada kasus angkutan kota, mobil angkutan kota adalah objek yang bergerak yang artinya butuh GPS *tracker* tertempel pada mobil untuk membuat *positioning augmented reality*. Untuk membuat penelitian ini tidak memakan banyak biaya, metoda ini tidak dapat digunakan karena akan memakan banyak biaya untuk membeli GPS *tracker* untuk setiap angkutan kota di kota Bandung. *Augmented reality* yang menggunakan marker juga sama kasusnya dengan *positioning augmented reality*. Memaksa untuk menggambar marker pada bodi mobil juga akan memakan banyak biaya. Metoda yang lebih masuk akal adalah *markerless augmented reality* yang hanya bergantung pada corak cat angkutan kota yang sudah ada sebelum penelitian ini dimulai.

Beberapa penelitian terkait yang mirip dengan penelitian ini adalah pengenalan wajah menggunakan metoda yang sama. Pada jurnal –*Smart Attendance Using Real Time Face Recognition (Smart – FR)*‖ oleh J.G. Rosgan Tharanga (2013) membahas tentang pengenalan karyawan dengan mendeteksi dan mengenali wajahnya untuk menandai presensi. Pada jurnal tersebut Haar *cascade* digunakan untuk mendeteksi wajah dan *Principal Component Analysis* digunakan sebagai metoda ekstraksi ciri sekaligus sebagai metoda pengenalan wajah yang menghasilkan ketepatan 68%. Jurnal lain tentang pengenalan wajah adalah –*Face Recognition using Neural Network*‖ oleh P.Latha, Dr.LGanesan dan Dr.S.Annadurai. Jurnal tersebut membahas tentang pengenalan wajah dengan foto sebagai input. Metoda yang digunakan adalah jaringan syaraf tiruan *Back Propagation* yang mencapai akurasi lebih dari 90%. Seperti dijelaskan pada jurnal tersebut, –In this paper, Face recognition using Eigen faces has been shown to be accurate and fast. When BPNN technique is combined with PCA, non linear face images can be recognized easily. Hence it is concluded that this method has the acceptance ratio is more than 90 % and execution time of only few seconds.‖

3. METODA DAN IMPLEMENTASI

A. Deteksi Objek

Deteksi objek dibutuhkan untuk mendeteksi objek tertentu yang tertangkap oleh kamera *smartphone* Android dan lalu memotong gambar sehingga hanya objek yang terdeteksi lah yang tersisi pada gambar. Gambar yang dipotong kemudian melewati proses ekstraksi ciri, dan keluarannya dipakai sebagai input dari proses pengenalan objek.

Haar *cascade* digunakan sebagai metoda deteksi objek. Metoda ini sangat efektif karena metoda ini sangat cepat dan memiliki tingkat kesalahan yang rendah. Deteksi objek pada aplikasi ini dibantu oleh *library* OpenCV untuk *smartphone* Android dan juga untuk PC.

Haar-like *feature* adalah fitur citra digital yang digunakan pada pengenalan objek. Inti dari deteksi objek dengan Haar *classifier* adalah Haar-like *feature*[6]. Fitur ini dari pada menggunakan intensitas nilai pixel, justru menggunakan perubahan nilai kontras antara kelompok pixel berbentuk persegi yang berdekatan. Perbedaan kontras dari kelompok pixel tersebut digunakan untuk menentukan gelap dan terangnya suatu area. Dua atau tiga kelompok yang berdekatan dengan kontras yang relatif berbeda membentuk sebuah Haar-like *feature*.

Untuk melatih Haar *classifier* adalah dengan mengikuti langkah-langkah berikut:

1. Kumpulkan sampel. Ada dua macam sampel yang dibutuhkan untuk melatih Haar *classifier*. 1000 atau lebih sampel positif dan 2000 atau lebih sampel negatif. Sampel positif adalah citra yang didalamnya terdapat objek yang menjadi target pelatihan. Sampel negatif adalah citra yang didalamnya tidak terdapat objek yang menjadi target tetapi merupakan latar dimana objek biasa berada. Untuk kasus angkutan kota, sampel positif adalah foto angkutan kota dan sampel negatif adalah foto jalan.

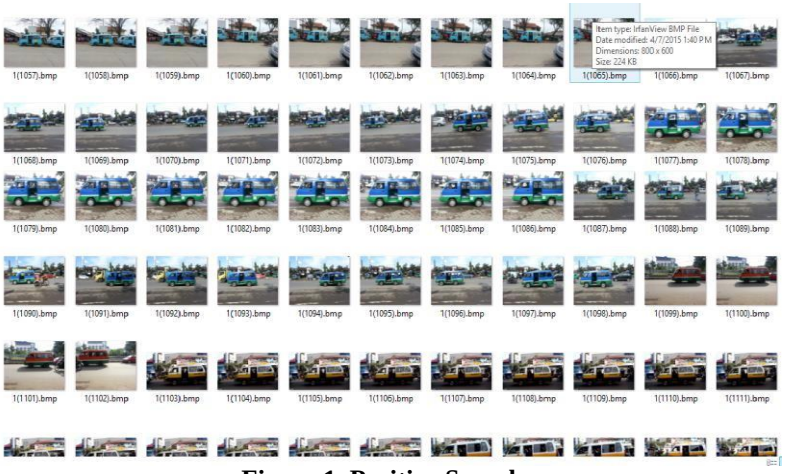

Figure 1. Positive Samples

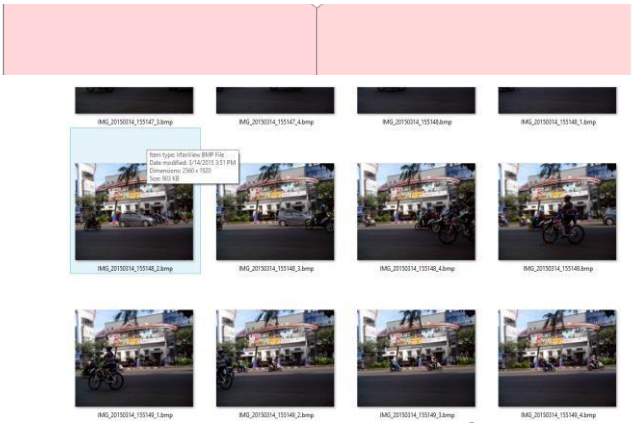

Figure 2. Negative Samples

2. Memotong atau mencatat koordinat dari objek target yang ada pada sampel positif dan membuat vektor dari sampel dengan menggunakan aplikasi yang disediakan oleh *library* OpenCV.
3. Melatih Haar *cascade classifier*. Haar *classifier* dapat dilatih dengan menggunakan aplikasi yang disediakan oleh *library* OpenCV.

Hasil keluaran dari proses pelatihan adalah berupa file .xml. File ini dapat digunakan oleh program dengan menggunakan *library* OpenCV atau JavaCV.

Tujuan dari dari porses deteksi objek ini adalah unuk menemukan angkutan kota pada frame yang tertangkap oleh kamera dan memotongnya sehingga hanya menyisakan gambar angkutan kota saja. Frame yang dipotong tersebut kemudian menjadi masukan untuk jaringan syaraf tiruan untuk proses pengenalan objek. Gambar yang dipotong secara otomatis diubah ukurannya menjadi 50 x 27. Angka ini merupakan hasil penghitungan ukuran rata-rata angkutan kota yang tertangkap oleh proses deteksi objek (tanpa mengalami perubahan ukuran) dari berbagai jarak pengambilan gambar. Dari rata-rata tersebut bisa didapatkan rasio panjang dan lebar dari citra yang dipotong.

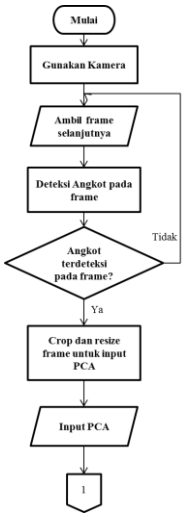

Figure 3. Object Detection Flow Chart

B. Rekognisi Objek

Rekognisi objek atau pengenalan objek memerlukan ekstraksi ciri dan jaringan syaraf tiruan (JST). Pada jurnal ini, *Principal Component Analysis* (PCA) digunakan sebagai metoda ekstraksi ciri, dan jaringan syaraf tiruan *Back Propagation* sebagai metoda pengenalan objek.

1. Principal Component Analysis

Principal Component Analysis (PCA) adalah teknik yang digunakan secara luas untuk aplikasi seperti pengecilan dimensi, kompresi data *lossy*, ekstraksi ciri, dan visualisasi data. PCA juga dikenal sebagai *the Karhunen-Loève transform* (Christopher M. Bishop F.R.Eng., 2007).

Metoda yang digunakan pada PCA adalah sebagai berikut:

- Kumpulkan beberapa data, pada kasus ini data yang dikumpulkan adalah foto angkutan kota yang ada di kota Bandung. Potong foto tersebut sehingga hanya menyisakan angkutan kota saja. Kemudian ubah ukuran seluruh foto menjadi ukuran (m x n) pixel.
- Lakukan pengurangan niali pixel terhadap rata-rata untuk setiap dimensi citra. Contohnya $X_{0,0}$ adalah nilai pixel pada koordinat (0,0) dari citra ke n dan $\bar{X}$ adalah rata-rata dari X_n . Lalu kurangkan hasil rata-rata dengan nilai dari setiap pixel. Contohnya: $(X_{0,0} - X_n)$.
- Hitung matriks kovarian untuk semua data
- Hitung eigenvalue dari matriks kovarian dengan menghitung deteminan dari matriks kovarian yang dikurangi oleh matriks identitas dikurang lambda sama dengan nol. $(A - \lambda I) = 0.$
- Hitung eigenvector dengan menyelesaikan persamaan $(A - \lambda I) = 0.$
- Urutkan eigenvector sesuai eigenvalue yang berkaitan muali dari yang terbesar hingga yang terkecil.
- Pilih sebanyak n buah eigenvector dengan eigenvalue yang tertinggi untuk membentuk vektor ciri. Vektor ciri ini dapat dikalikan dengan matriks citra untuk mengkompres citra menjadi n dimensi dan menggunakannya menjadi input untuk *Back Propagation*..

2. Jaringan Syaraf Tiruan *Back Propagation*

Jaringan syaraf tiruan adalah prosesor yang terdistribusi secara parallel dan massive yang memiliki kecendrungan alami untuk menyimpan pengetahuan dari pengalaman dan membuatnya bisa digunakan. Hal itu menyerupai otak dalam dua aspek:

- Pengetahuan didapatkan oleh jaringan melalui proses belajar.
- Kekuatan hubungan antar neuron dikenal sebagai bobot sinaptik digunakan untuk menyimpan pengetahuan. (Haykin, Simon. 1994).

Back Propagation adalah teknik khusus untuk menerapkan *gradient descent* dalam *weight space* untuk jaringan *multilayer feedforward*. Ide dasar dari teknik tersebut adalah untuk dapat menghitung secara efisien *partial derivatives* dari sebuah fungsi perkiraan F(w:x) yang diwujudkan oleh jaringan terhadap semua elemen vektor bobot yang dapat diatur w untuk nilai yang diberikan kepada vektor *input* x (Haykin, Simon 1994)

Seperti terlihat pada figure 4, *Back Propagation* terdiri atas 3 jenis *layer* neuron, *input layer*, *hidden layer*, *output layer*. *Back Propagation* membutuhkan *input* dan target. Tujuannya adalah untuk membuat *output* yang dihasilkan memiliki nilai yang sedekat mungkin dengan target. *Input layer* menerima sejumlah informasi, *output layer* akan membandingkan hasil keluaran dengan target, dan *hidden neuron* akan menyesuaikan diri untuk membuat perbedaan antara keluaran dan target sekecil mungkin.

Input untuk *hidden layer* adalah jumlah dari nilai *input* yang telah dikalikan dengan bobot. Persamaan [7]:

$$Sum_m = \sum_{j=1}^{i} X_j U_{mj}$$

Keluaran dari *hidden layer* adalah hasi dari penjumlahan *input* di atas setelah melewati fungsi aktifasi. Persamaan [7]:

$$Z_m = \frac{1}{1 + Exp(-Sum_m)}$$

Input dan *output* dari *output layer* memiliki persamaan yang mirip dengan *hidden layer*. Jumlah dari nilai keluaran *hidden output* yang telah dikalikan dengan bobotnya masing-masing kemudian melewati fungsi aktifasi. [7]

$$Sum_m = \sum_{j=1}^{i} Z_j W_{mj}$$

$$Y_m = \frac{1}{1 + Exp(-Sum_m)}$$

Untuk menghitung kesalahan dari *output* yang sebenarnya dapat dilakukan dengan menghitung perbedaan antara *output* dan target. Persamaan [7]:

$$EY_i = T_i - Y_i$$

Dimana *EY* mewakili *error* dari *output* yang sebenarnya, T_i mewakili target *output* di neuron *i*, and O_i mewakili *output* yang sebenarnya di neuron *i*. Untuk menghitung *error* pada *hidden layer* dapat dilakukan dengan persamaan sebagai berikut [7]:

$$Sum = \sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}\, EY_j$$

$$EZ_j = (Sum)Z_j(1 - Z_j)$$

Dimana W_{ij} mewakili bobot antara *hidden layer* di neuron *i* dan *output layer* di neuron *j*. EZ_j mewakili kesalahan dari Z_j, *output* di *hidden layer* neuron *j*. Untuk memperbarui bobot dapat dilakukan dengan menjumlahkan hasil koreksi kesalahan dengan bobot yang bersangkutan. Persamaan [7]:

$$\Delta_i^W = Z_i EY_i \alpha$$

$$W_{ij} = W_{ij} + \Delta_i^W$$

Dimana Δ_i^W mewakili koreksi *error* dari bobot antara *hidden layer* dan *output layer*, dan α adalah *learning rate*. Untuk memperbarui bobot antara *input layer* dan *hidden layer* dapat dilakukan dengan menggunakan persamaan yang mirip [7]:

$$\Delta_i^U = X_i EZ_i \alpha$$

$$U_{ij} = U_{ij} + \Delta_i^U$$

C. Hasil

Hasil dari implementasi metode dapat dilihat berdasarkan pengujian yang dilakukan terhadap sistem. Pengujian tersebut diantaranya adalah pengujian akurasi terhadap gambar angkutan kota yang ada di monitor komputer, pengujian waktu proses saat ada satu atau lebih angkutan kota yang tertangkap kamera, dan pengujian terhadap angkutan kota asli di jalan atau terminal.

Berikut adalah hasil pengujian akurasi:

Dengan menghitung rata-rata hit-rate dari tabel di atas didapatkan total akurasi dengan mengubahnya ke dalam persen. Seperti dapat di lihat pada tabel di bawah, akurasi dari sistem mencapai 40%.

Table 1 Tabel persentase akurasi

Rata-rata *hit-rate*	4.072866
Akurasi	40.73%

Berikut adalah pengujian waktu proses:

Pengujian ini bertujuan untuk menghitung waktu rata-rata yang dibutuhkan oleh sistem dalam melakukan proses dari mengambil frame hingga memunculkan informasi angkutan kota. Pengujian dilakukan dengan mengambil total waktu dari beberapa sampel pengujian. Sampel didapatkan dengan menggunakan untuk memunculkan informasi dari angkutan kota yang ada pada foto di bawah. Waktu di dapatkan dengan menuliskan kode untuk memunculkan selisih waktu pada aplikasi yang dimunculkan setiap kali aplikasi memunculkan informasi seperti pada gambar.

Berikut adalah tabel hasil pengukuran waktu proses:

Tabel 2: Tabel pengukuran waktu proses

	Lama Proses (dalam second)			
	1 Angkot	2 Angkot	3 Angkot	4 Angkot
Rata-rata	0.57845	0.6027	1.09855	1.02595

Berikut adalah pengujian akurasi sistem terhadap angkutan kota yang sebenarnya:

Tujuan pengujian adalah untuk menguji ketepatan informasi ketika aplikasi di gunakan terhadap angkutan kota yang ada di jalan atau terminal. Pengujian tidak dilakukan terhadap semua angkutan kota yang ada di kota Bandung, melainkan hanya beberpa angkutan kota saja. Pengujian dilakukan dengan cara menggunakan aplikasi di suatu tempat di mana ada angkutan kota yang sering berhenti. Aplikasi secara otomatis mencatat hasil identifikasinya untuk memudahkan pengujian. Pengujian dilakukan selama 1 jam dengan 2 angkutan kota yang berbeda. Hasil pengujian adalah seberapa banyak informasi yang benar selama proses identifikasi. Pengujian dilakukan pada angkutan kota Cicaheum – Ciwastra – Derwati dan Cijerah – Ciwastra – Derwati karena keduannya secara kebetulan berhenti dan lewat di daerah yang sama. Berikut adalah hasil dari pengujian:

1. Percobaan pada angkutan kota yang tidak bergerak

Tabel 3: Pengujian terhadap angkutan kota tidak bergerak

No	Nama Angkot	Total *Frame*	*Output* benar
1	Cicaheum - Ciwastra - Derwati	407	332
2	Cijerah - Ciwastra - Derwati	95	34

Sebanyak 407 *frame* angkutan kota Cicaheum – Ciwastra – Derwati terdeteksi, didapatkan 332 kali mengeluarkan *output* benar. Pada angkutan kota lain yaitu Cijerah – Ciwastra – Derwati yang dalam keadaan tidak bergerak, dari 95 kali frame yang terdeteksi, terdapat 34 *output* yang benar. Aplikasi juga tidak akan bekerja dengan benar pada angkutan kota yang tidak mirip dengan foto angkutan kota yang digunakan saat training, baik berbeda kontrasi warna maupun berbeda corak.

2. Percobaan pada angkutan kota yang bergerak (Cicaheum – Ciwastra – Derwati)
   Pada angkutan kota yang bergerak didapatkan 50 kali angkutan kota terdeteksi. Dari 50 frame tersebut hanya mampu mengeluarkan 9 *output* yang benar. Hal ini karena pengaruh angkutan kota yang bergerak membuat angkutan kota yang dideteksi menjadi buram saat di crop atau miring.

4. KESIMPULAN DAN SARAN

Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

3. Aplikasi mampu mengidentifikasi angkutan kota, akan tetapi hanya dengan ketepatan 40% dalam keadaan *smartphone* dan angkutan kota tidak bergerak dan tidak ada kemiringan sama sekali..
4. Aplikasi mampu menampilkan informasi dengan menggabungkan objek virtual berupa tulisan dan objek nyata berupa angkutan kota yang tertangkap oleh kamera smartphone dengan rata-rata waktu yang di butuhkan dari proses deteksi hingga menampilkan informasi 0.57 detik
5. Aplikasi kurang handal dalam mengenali angkutan kota berwarna hijau dan biru dikarenakan terlalu banyak angkutan kota yang berwarna hijau dan biru. Aplikasi lebih handal dalm mengenali angkutan kota berwarna biru muda, merah, coklat, putih, dan kuning.

Adapun saran untuk pengembangan lebih lanjut adalah:

1. Mengganti metoda ekstraksi ciri dengan yang lebih mendukung warna rgb.
2. Membuat objek 2D ataupun 3D dari angkutan kota sebagai informasi tambahan, dan agar lebih menarik
3. Sebaiknya untuk setiap data training tidak mengambil foto dari objek yang sama berulang-ulang agar hasil lebih maksimal

REFERENSI

[1] Kim, Young-geun & Kim, Won-jung (2014). ‖Implementation of Augmented Reality System for Smartphone Advertisements‖. International Journal of Multimedia and Ubiquitous Engineering. 9 (2), 385-392.
[2] OpenCV Developers Team. About OpenCV [online]. Tersedia: http://opencv.org/about.html [26 May 2015].
[3] OpenCV Developer Team. Face Detection using Haar Cascade [online]. Tersedia: http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html [26 May 2015]
[4] Smith, I. Lindsay (2002). ‖A tutorial on Principal Components Analysis‖.
[5] Viola, Paul & Jones, J. Michael (2001) ‖Rapid Object Detection using a Boosted Cascade of Simple Features‖. Computer Vision and Pattern Recognition.
[6] Wilson, Phillip & Fernandez, John (2006). ‖Facial Feature Detection Using Haar Classifiers‖. JCSC. 21 (4).
[7] Latha, P., et al (2011). ‖Face Recognition using Neural Networks‖.