

# EKSTRAKSI FITUR MIRIP HAAR UNTUK PENDETEKSIAN OBYEK PADA REAL-TIME VIDEO MENGGUNAKAN OPENCV DAN IMPLEMENTASINYA PADA SISTEM TERTANAM BEAGLEBOARD XM

Zarnuji Sidiq<sup>1</sup>

<sup>1</sup>Jurusan Fisika FMIPA Univ. Brawijaya  
Email:0910930070@mail.uib.ac.id

## Abstrak

Robot memerlukan aksi yang tepat untuk menanggapi suatu keadaan lingkungan. Kondisi lingkungan dapat diperoleh dari sensor-sensor yang terhubung dengan suatu unit pemroses, salah satu sensor yang dapat digunakan adalah sensor visual. Aksi terhadap obyek yang hanya dapat dikenali secara visual sangat bergantung pada kemampuan robot dalam mengolah informasi warna atau bentuk obyek. Dalam penelitian ini digunakan pendeteksian obyek berdasarkan ekstraksi fitur yang banyak diterapkan pada pengenalan wajah otomatis. Fitur-fiturnya kemudian diolah menggunakan metode pembelajaran terarah (*supervised learning*) *AdaBoost* untuk melakukan pembelajaran terhadap suatu obyek tertentu. Tahap pembelajaran dilakukan menggunakan rasio sampel positif dan negatif sebesar 5:3 untuk membuat tabel *look-up*. Tiga set tabel *training* dibuat dengan jumlah sampel dan *stage training* yang berbeda. Tiga set tabel yang dibuat telah diuji menggunakan 2 buah berkas video suatu lingkungan yang diambil dari dua sudut pandang yang berbeda. Telah didapatkan dalam penelitian ini rasio pendeteksian pada kedua video memberikan kecenderungan yang sama. Suatu aturan yang sederhana ditunjukkan dalam penelitian ini untuk meningkatkan akurasi tabel *look-up*. Pustaka OpenCV digunakan dalam penelitian ini karena kemampuannya yang dapat ditanamkan pada bermacam-macam perangkat keras, termasuk piranti-piranti bergerak dan sistem tertanam. Kecepatan pendeteksian obyek menggunakan sistem tertanam *Beagleboard XM* juga dijelaskan dalam penelitian ini.

Kata kunci :OpenCV, fitur-mirip-haar, *computer vision*, pengenalan obyek

## Pendahuluan

Perkembangan teknologi di era modern ini semakin mempertegas kebutuhan orang akan sistem-sistem yang dapat bekerja secara mandiri, salah satunya adalah robot. Robot dibuat dan diprogram supaya dapat secara mandiri melakukan eksekusi tertentu terhadap kondisi lingkungan. Kemampuan mandiri ini dapat dicapai dengan kecerdasan buatan (*artificial intelligence*) atau pembelajaran mandiri (*adaptive learning*). Salah satu contoh pemanfaatan robot yang menggunakan teknik ini adalah robot pembersih meja makan [1]. Kebanyakan robot-robot cerdas menggunakan sensor visual untuk mendapatkan informasi kondisi lingkungan. Sensor visual merupakan *sensing element* yang *powerful* terutama untuk menentukan eksekusi pergerakan robot terhadap suatu obyek visual tertentu [2].

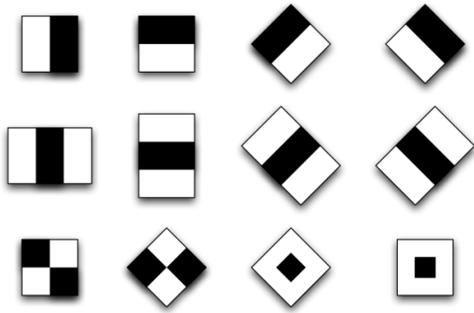
Teknik-teknik pengenalan obyek berbasis informasi *visual* telah banyak dikembangkan dan dipelajari. Beberapa diantaranya menggunakan informasi warna sebagai informasi utama. Metode lain menggunakan pendekatan bentuk dasar obyek yang akan dideteksi. Keperluan yang lebih rumit, misalnya menentukan jarak perpindahan obyek dari satu titik ke titik yang lain memerlukan

teknik tambahan untuk melakukan *tracking*, misalnya menggunakan *kernel* dari histogram [3].

Teori multiresolusi yang dijelaskan oleh [4] digunakan sebagai dasar konsistensi fitur ketika obyek yang dideteksi memiliki ukuran yang berbeda. Fitur-fitur mirip Haar [5, 6] sebagai *feature extractor* digunakan sebagai akibat langsung dari teori multiresolusi. Gambar 1 menunjukkan fitur-fitur yang digunakan dalam *paper* ini.

Dasar algoritma pengenalan wajah yang ditawarkan oleh Viola dan Jones [7] digunakan dalam penelitian ini. Pustaka OpenCV menerapkan algoritma ini untuk melakukan pendeteksian wajah berbasis *cascade*. *Boosting* [8, 9] digunakan untuk melakukan *training* fitur-fitur obyek. *Gentle AdaBoost* yang ditawarkan oleh [10] dan dibuktikan oleh [5] memiliki performa *error-rate* yang paling kecil [11] dibandingkan turunan *AdaBoost* yang lain.

Metode *boosted-cascade classifier* seperti digunakan oleh [7] diterapkan pada fasa *training* obyek. Pengaruh karakter dataset, jumlah sampel, parameter-parameter *boosting*, serta jumlah *stage* dalam fasa *training* dikaji dalam penelitian ini. Teknik sederhana untuk menentukan kriteria data-set yang baik untuk digunakan dalam fasa *training* juga dijelaskan. Hasil *training* kemudian diterapkan dan diuji pada sistem tertanam *Beagleboard XM*.



Gambar 1. Fitur-fitur mirip Haar [5, 6]

## Metode

Dataset [12-14] digunakan sebagai sampel positif dan sampel negatif pada fasa *training*. Gambar 2 adalah beberapa contoh dataset yang digunakan. Tiga set tabel *look-up* dibuat menggunakan jumlah sampel yang berbeda namun memiliki perbandingan jumlah sampel positif dan sampel negatif 5:3. Masing-masing SET dibagi menjadi 3 buah subset dengan 3 kali pengubahan jumlah *round* dalam fasa *training*, yaitu 10x, 20x, dan 30x. Parameter-parameter pembuatan tabel *look-up* ditunjukkan dalam tabel 1.



Gambar 2. Contoh sampel negatif (baris atas) dan sampel positif (baris bawah) yang digunakan.

Tabel 1. Parameter pembuatan SET tabel *look-up*

No.	Aspek	Nilai
1.	Jumlah sampel positif(I, II, III)	(100, 199, 299) buah
2.	Ukuran sampel positif (lebar×tinggi)	128×128px
3.	Jumlah sampel negatif(I, II, III)	(60, 120, 180) buah
4.	Ukuran sampel negatif (lebar×tinggi)	(1024×768)px
5.	<i>Hit rate</i> minimal	0.999
6.	<i>False Alarm Rate</i> maksimal	0.5
7.	Ukuran Minimum <i>Bounding Box</i>	(30×30)px

Perangkat keras yang digunakan dalam penelitian ini dibagi menjadi 2, masing-masing untuk *training* dataset serta untuk pengujian program. Pembuatan tabel *look-up* menggunakan sebuah laptop dengan CPU Intel Core i3, dan kartu grafis ATI Mobility Radeon 5470 HD. Pengujian program menggunakan *system on chip* Beagleboard XM dengan pemroses DM3730 dan dukungan *hardware floating point* dan frekuensi 1GHz.

Beagleboard XM memiliki pemroses *parallel TI* DM3730 dan TMS320C64xx. Pengujian algoritma dan tabel dilakukan menggunakan DM3730.

Dua berkas video masing-masing dengan ukuran *frame* 640×480 piksel digunakan untuk menguji tiga set tabel *look-up* yang dibuat. Kedua berkas video ini selanjutnya disebut *Sarinah[A]* dan *Sarinah[B]* dan memiliki perbedaan sudut pandang pengambilan gambar. Pendeteksian dilakukan tiap 25 *frame* untuk setiap berkas video. Contoh *frame* dua berkas video ini ditunjukkan dalam gambar 3. Metode yang digunakan oleh [15] digunakan untuk menentukan *recognition rate* dan *false detection rate* algoritma dan tabel *look-up* yang telah dibuat.

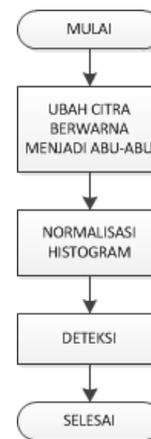


Gambar 3. Kiri: *Sarinah[A]*. Kanan: *Sarinah[B]*

Tahap pra-pemrosesan menggunakan normalisasi histogram diperlukan sebelum obyek perhatian dalam citra satu warna dicari. Penghitungan normalisasi histogram dilakukan menggunakan *cumulative distribution function* yang ditunjukkan dalam persamaan (1). Keseluruhan algoritma ditunjukkan dalam diagram alir pada gambar 4.

$$H'_i = \sum_{0 \leq j < i} H_{i_{gray}}(j) \quad (1)$$

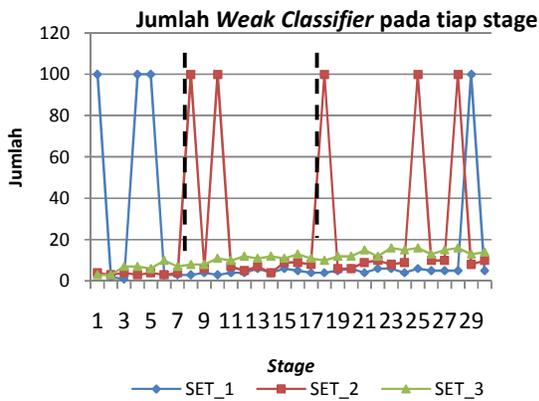
Diagram alir tahap pra pemrosesan serta tahap pemrosesan dan pendeteksian obyek dalam tiap *frame* video ditunjukkan dalam gambar 4.



Gambar 4. Diagram alir pendeteksian obyek

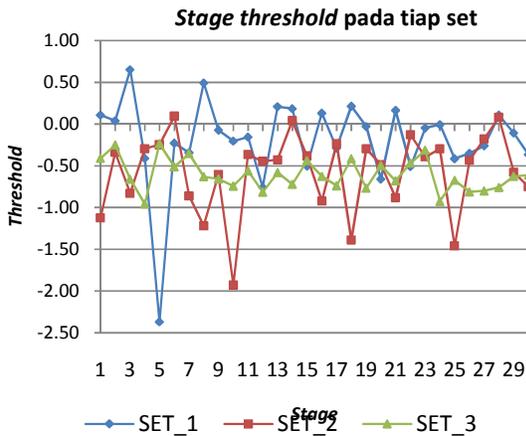
## Hasil dan Pembahasan

Tiga SET tabel *look-up* memiliki jumlah pengklasifikasi lemah pada tiap *stage* pada fasa *training* ditunjukkan dalam gambar 5.



Gambar 5. Jumlah pengklasifikasi tiap *stage* pada tiap SET

Jumlah *weak classifier* pada SET\_1 dan SET\_2 memiliki variasi nilai yang berfluktuasi yang signifikan. SET\_3 menunjukkan jumlah pengklasifikasi lemah pada tiap *stage* tidak mengalami banyak perubahan.

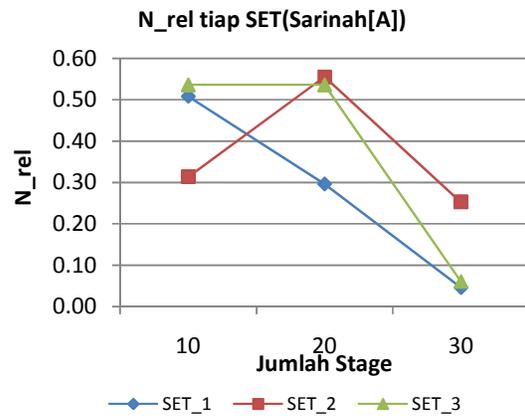


Gambar 6. Nilai ambang *decision tree* tiap *stage* pada masing-masing SET

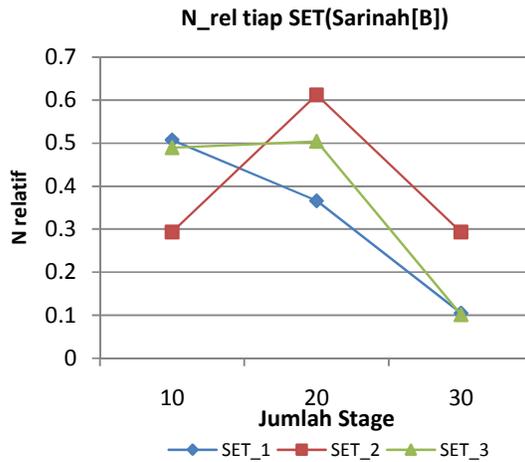
Analisis menggunakan [15](persamaan (2)) pada seluruh SET menunjukkan pengaruh kualitatif yang signifikan akibat tingkat kestabilan nilai ambang pada tiap *round* dan jumlah pengklasifikasi lemah tiap *stage* pada masing-masing SET terhadap tingkat akurasi pendeteksian. Tiga SET tabel *look-up* yang diuji dengan 2 berkas video memberikan nilai-nilai  $R_{rel}$  dan  $N_{rel}$  dengan kecenderungan (*trend*) yang sama. Kecenderungan ini ditunjukkan pada gambar 7 dan 8.

$$R_{rel} = \frac{n_{positif}}{N_{box}} \quad (2)$$

$$N_{rel} = \frac{n_{positif}}{N_{aktual}}$$

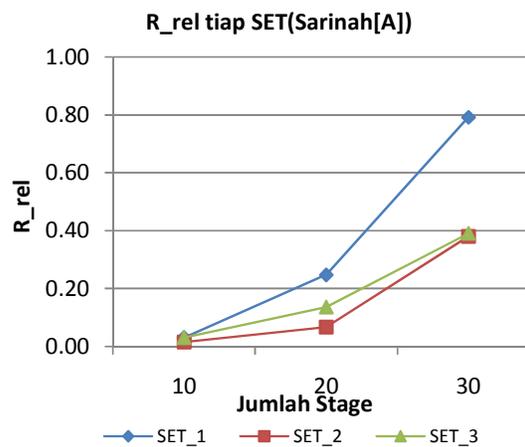


Gambar 7.  $N_{rel}$  tiap SET untuk Sarinah[A]



Gambar 8.  $N_{rel}$  tiap SET untuk Sarinah[B]

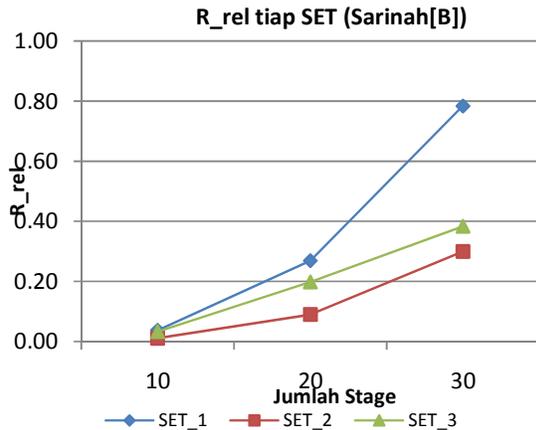
Gambar 9 dan 10 menunjukkan nilai  $R_{rel}$  pada tiap SET. SET\_2 menunjukkan nilai  $R_{rel}$  paling tinggi dihasilkan dari tabel *look-up* hasil *training* dengan 30 *stage training*.



Gambar 9.  $R_{rel}$  tiap SET untuk Sarinah[A]

Tabel OPT dibuat dengan mempertimbangkan performa pendeteksian tiap SET tabel *look-up* yang telah dibuat. Nilai  $N_{rel}$  tertinggi pada masing-masing SET adalah pada 20x *training stage*. Sedangkan nilai  $R_{rel}$  pada masing-masing

berkas video uji adalah ketika SET tabel dibuat dengan 30x *training stage*. Nilai  $R_{rel}$  dan  $R_{rel}$  tertinggi masing-masing digunakan sebagai pertimbangan jumlah *stage* dan banyaknya sampel obyek dalam fasa *training*. Parameter-parameter yang digunakan dalam pembuatan tabel OPT ini ditunjukkan pada tabel 2. Beberapa hasil pendeteksian menggunakan tabel OPT ditunjukkan pada gambar 11.



Gambar 10.  $R_{rel}$  tiap SET untuk Sarinah[B]

Hasil pendeteksian menggunakan tabel OPT menunjukkan adanya perubahan kemampuan pendeteksian yang sangat drastis dibandingkan hasil pendeteksian yang dapat dicapai menggunakan 3 SET tabel *look-up* yang telah dibuat sebelumnya.

Tabel 2. Parameter pembuatan tabel OPT

No.	Parameter	Nilai
1	Besar obyek terkecil	40x40 px
2	<i>Hit-rate</i>	0.999
3	<i>False alarm rate</i>	0.5
4	Jumlah <i>stage</i>	20
5	Jumlah sampel positif	800
6	Jumlah sampel negatif	600



Gambar 11. Hasil pendeteksian tabel OPT

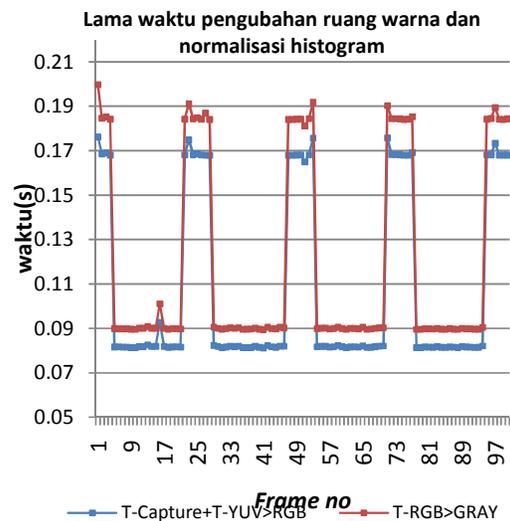
Tabel 3 merupakan hasil pendeteksian obyek mobil menggunakan tabel OPT. Nilai-nilai  $R_{rel}$  pada masing-masing berkas video menunjukkan perubahan yang cukup signifikan dibandingkan  $R_{rel}$  menggunakan 3 SET tabel yang telah dibuat sebelumnya dengan selisih mencapai 0.2. Perbedaan nilai  $R_{rel}$  yang cukup signifikan dapat merupakan dampak dari adanya obyek lain yang

memiliki bentuk yang menyerupai obyek perhatian. Nilai  $N_{rel}$  juga menunjukkan kenaikan penurunan nilai yang signifikan mencapai 0.4.

Tabel 3. Hasil pendeteksian menggunakan tabel OPT

	Sarinah[A]	Sarinah[B]
<b>N_Obj</b>	425	271
<b>N_Pos</b>	222	222
<b>N_Box</b>	289	305
<b>N_FalsePos</b>	67	154
<b>R_Rel</b>	0.77	0.49
<b>N_Rel</b>	0.52	0.56

Penerapan algoritma dan tabel *look-up* OPT telah diuji menggunakan perangkat Beagleboard XM. Kecepatan eksekusi algoritma dibandingkan menjadi 3, kecepatan konversi ruang warna RGB menjadi skala abu-abu, kecepatan normalisasi histogram, serta kecepatan pendeteksian obyek. Kecepatan 3 algoritma ini ditunjukkan pada gambar 10 dan 11.

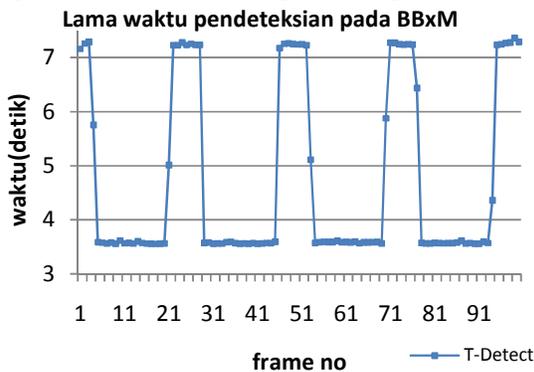


Gambar 12. Waktu yang diperlukan untuk mengubah ruang warna dan normalisasi warna pada BBxM

Grafik perubahan kecepatan eksekusi algoritma pada gambar 12 dan 13 menunjukkan perubahan waktu eksekusi antara 3 hingga 8 detik untuk melakukan pendeteksian. Sekitar 13-17 *frame* pendeteksian berjalan selama lebih kurang 3~5 detik. Sedangkan 6-8 *frame* algoritma berjalan dengan kecepatan 7~8 detik. Hal ini terjadi secara periodik sepanjang video uji.

Kecepatan eksekusi algoritma seperti dijelaskan sebelumnya disebabkan karena dua hal utama. Penyebab pertama unit pemroses memiliki kecenderungan untuk melakukan *clean-up* memori-memori *buffer* yang tidak terpakai lagi pada siklus pendeteksian selanjutnya. Kedua, OpenCV adalah pustaka sumber terbuka yang dioptimalkan

untuk *processor* berbasis arsitektur Intel yang membutuhkan sumber daya energy yang besar. Alasan ini juga telah dikemukakan oleh [11]. Kecepatan eksekusi pada pemroses *mobile* (terutama dengan arsitektur ARM) dengan sumber daya energi minimal tidak dapat diharapkan terlalu banyak jika algoritma yang digunakan murni berbasis pustaka OpenCV.



Gambar 13. Waktu pendeteksian obyek pada Beagleboard XM

## Simpulan

Dalam penelitian ini, fitur-fitur yang diekstraksi menggunakan fungsi wavelet dapat dimanfaatkan untuk pendeteksian obyek secara mandiri menggunakan algoritma *AdaBoost*. Dengan menggunakan algoritma *AdaBoost* dapat diketahui bahwa nilai

ambang (*threshold*) tiap *stage* dalam tabel *look-up* yang terlalu rendah serta jumlah pengklasifikasi lemah yang tidak stabil pada tiap *stage* memberikan hasil pendeteksian ( $R_{rel}$  dan  $N_{rel}$ ) yang buruk. Satu aturan sederhana dapat digunakan untuk memperkuat kemampuan tabel *look-up* dalam pendeteksian obyek berdasarkan kemampuan tabel *look-up* yang telah dibuat sebelumnya, yaitu dengan mempertimbangkan nilai  $N_{rel}$  dan  $R_{rel}$  pada tiap-tiap tabel. Tabel *look-up* dapat dikatakan baik jika nilai  $R_{rel}$  dan  $N_{rel}$  mendekati satu. Selain nilai  $R_{rel}$  dan  $N_{rel}$  harga ambang yang semakin stabil atau bergerak dengan kemiringan perubahan yang tidak terlalu cepat akan memberikan hasil pendeteksian yang lebih baik.

Penerapan *computer vision* pada sistem tertanam memberikan kecepatan proses 3 detik hingga 7 detik pada berkas video dengan ukuran *frame* 640x480. Penggunaan FPGA atau DSP dalam pengolahan citra dapat dimanfaatkan untuk meningkatkan kecepatan proses pendeteksian. Tahapan pembelajaran menggunakan metode *machine learning* yang lain merupakan salah satu kajian yang menarik dalam bidang *computer vision*. Penggunaan data *raw* dari kamera dapat digunakan untuk mencapai kecepatan proses yang lebih baik.

## Daftar Pustaka

- [1] M. Attamini, T. Araki, T. Nakamura, T. Nagai. Visual Recognition System for Cleaning Tasks by Humanoid Robots. dalam *International Journal of Advanced Robotic Systems*. 10: 384. 2013.
- [2] M.S. Guzel, R. Bicker. Visual Based Obstacle Avoidance Techniques. InTech. 2011.
- [3] P.P. Dash, D. patra, S.K. Mishra. Kernel-based Object Tracking using Color Histogram Technique. dalam *Int. Journal of Electronics and Electrical Engineering*. 2: 28-35. 2012.
- [4] S.G. Mallat. A Theory for Multiresolution Signal Decomposition: A Wavelet Representation. dalam *IEEE Trans. Pattern Analysis and Machine Intelligent*. 11. 1989.
- [5] R. Lienhart, A. Kuranov, V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. dalam *MRL Technical Report*. 2002.
- [6] P. Viola, M.J. Jones. Robust Real-Time Face Detection. dalam *International Journal of Computer Vision*. 57: 137-154. 2004.
- [7] P. Viola, M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. dalam *CVPR2001*. 2001.
- [8] Y. Freund, R.E. Schapire. Experiments with a New Boosting Algorithm. dalam *Machine Learning: Proceedings of Thirteenth International Conference*. 148-156. 1996.
- [9] R. Schapire, Boosting, dalam, Video Lectures, 2007, pp. Boosting, theory, algorithm.
- [10] J.H. Friedman, T. Hastie, R. Tibshirani. Adaptive Logistic Regression: A Statistical View of Boosting. dalam *The Annals of Statistics*. 2: 337-374. 2000.
- [11] G. Günlü. Embedded Face Detection and Recognition. dalam *International Journal of Advanced Robotic Systems*. 9. 2012.
- [12] G. Griffin, A. Holub, P. Perona. The Caltech 256. 2006.
- [13] P. Carbonetto, G. Dorkó. INRIA Car Data Set. 2008.
- [14] R. Fergus, P. Perona, A. Zisserman. Background Dataset. 2008.
- [15] P. Zhang, X. Guo. A Cascade Face Recognition System Using Hybrid Feature Extraction. dalam *Digital Signal Processing*. 22: 987-993. 2012.