

PERANGKAT LUNAK PENGENDALI *POINTER* MENGGUNAKAN PELACAKAN MATA (*EYE TRACKING*) DENGAN ALGORITMA LUCAS KANADE

Yuli Syarif^{*)}, R. Rizal Isnanto, and Kodrat Iman Satoto

Jurusan Teknik Elektro, Universitas Diponegoro Semarang
Jl. Prof. Sudharto, SH, kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)} *Email: yuli.syarif@hotmail.com*

Abstrak

Sistem pelacakan objek pada citra bergerak saat ini telah banyak diterapkan dan digunakan untuk berbagai tujuan khusus. Salah satu algoritma yang digunakan untuk melakukan pelacakan adalah algoritma Lucas-Kanade. Algoritma pelacakan Lucas-Kanade ini diusulkan pada tahun 1981 oleh Bruce D. Lucas dan Takeo Kanade. Algoritma ini mengasumsikan bahwa titik yang dilacak dan titik-titik tetangga dari titik yang dilacak tersebut memiliki nilai intensitas yang sama dari frame ke frame, serta perpindahan posisi titik yang dilacak dari frame ke frame adalah kecil. Penelitian ini dibuat untuk melakukan penelitian pada salah satu metode pelacakan yaitu algoritma Lucas-Kanade. Penelitian yang dilakukan adalah membuat perancangan dan implementasi algoritma Lucas-Kanade yang digunakan untuk melakukan pelacakan fitur pada mata, pengendali posisi pointer berdasarkan posisi fitur mata dan pengendali fungsi klik menggunakan kedipan mata. Pengujian yang dilakukan pada penelitian ini adalah mengukur besar kecepatan gerak pointer, keberhasilan melakukan klik dan keakuratan sistem. Berdasarkan hasil penelitian diperoleh hasil untuk rata-rata waktu pada skala gerak tinggi sebesar 5,9 detik, pada skala gerak normal sebesar 7,9 detik dan pada skala gerak rendah sebesar 9,7 detik, untuk tingkat keberhasilan pada webcam 16 MP sebesar 87,8% sedangkan pada webcam 0,3 MP sebesar 78,9% dan untuk presentase tingkat keberhasilan pada skala gerak tinggi sebesar 77,7% dengan rata-rata waktu sebesar 7,5 detik, presentase tingkat keberhasilan pada skala gerak normal sebesar 86,7% dengan rata-rata waktu sebesar 5,9 detik dan presentase tingkat keberhasilan pada skala gerak rendah sebesar 90% dengan rata-rata waktu sebesar 8,2 detik.

Kata Kunci : Lucas-Kanade, pelacakan mata, pointer, Viola-Jones, webcam

Abstract

System tracking moving objects in the image has been widely implemented and used for various special purposes. One algorithm is used to triangulate the Lucas-Kanade algorithm. Lucas-Kanade tracking algorithm was proposed in 1981 by Bruce D. Lucas and Takeo Kanade. This algorithm assumes that the tracked point and neighboring points of the tracked points have the same intensity values from frame to frame, and the position shift points tracked from frame to frame is small. The final assignment is made to do some research on one method of tracking the Lucas-Kanade algorithm. Research is to make the design and implementation of the Lucas-Kanade algorithm is used to perform the tracking feature on the eyes, controlling the pointer position by eye feature position and controlling click functions using eye blink. Tests were conducted In this final project is to measure the velocity of the pointer, clicking the success and accuracy of the system. Based on the research results for the average time on the high scale motion was 5.9 seconds, normal scale motion was 7.9 seconds and low scale motion was 9.7 seconds, success rate at 16 MP webcam was 87.8% while at 0.3 MP webcam was 78.9% and success rate on high scale motion was 77.7% with the average time 7.5 seconds, success rate on normal scale motion was 86.7% with average time 5.9 seconds success rate on a low scale motion was 90% with average time 8.2 seconds.

Keywords: eye tracking, Lucas-Kanade, pointer, Viola-Jones, webcam

1. Pendahuluan

Sistem pelacakan objek pada citra bergerak saat ini telah banyak diterapkan dan digunakan untuk berbagai tujuan

khusus. Salah satu algoritma yang digunakan untuk melakukan pelacakan adalah algoritma Lucas-Kanade. Algoritma pelacakan Lucas-Kanade ini diusulkan pada tahun 1981 oleh Bruce D. Lucas dan Takeo Kanade.

Algoritma ini mengasumsikan bahwa titik yang dilacak dan titik-titik tetangga dari titik yang dilacak tersebut memiliki nilai intensitas yang sama, serta perpindahan posisi titik yang dilacak dari *frame* ke *frame* adalah kecil. Posisi, arah gerak, besar pergerakan yang dilakukan dan kecepatan gerak dari titik yang dilacak pada *frame* yang selanjutnya dapat diukur menggunakan algoritma tersebut [6].

Penelitian sebelumnya tentang pelacakan mata menggunakan metode *haar cascade* telah dilakukan [9][12]. Penelitian tentang pelacakan mata menggunakan metode *Fisherface* juga telah dilakukan [11].

penelitian ini dibuat untuk melakukan penelitian pada salah satu metode pelacakan yaitu algoritma Lucas-Kanade. Penelitian yang dilakukan adalah membuat perancangan dan implementasi algoritma Lucas-Kanade yang digunakan untuk melakukan pelacakan fitur pada mata, pengendali posisi pointer berdasarkan posisi fitur mata dan pengendali fungsi klik menggunakan kedipan mata. Pengujian yang dilakukan pada penelitian ini adalah mengukur besar kecepatan gerak pointer, keberhasilan dan keakuratan dalam melakukan klik.

Tujuan dari penelitian ini adalah sebagai berikut.

1. Merancang dan membangun suatu sistem pelacakan fitur pada mata menggunakan algoritma Lucas-Kanade.
2. Merancang dan membangun suatu sistem pengendalian posisi *pointer* berdasarkan posisi mata.
3. Merancang dan membangun suatu sistem pengendalian fungsi klik menggunakan kedipan mata.

Batasan masalah dalam pembahasan penelitian ini meliputi :

1. Sistem hanya mampu melakukan pengendalian gerakan *pointer*, klik kanan, klik kiri dan *double* klik.
2. Pada proses pendeteksian wajah menggunakan data latih yang telah tersedia.
3. Sistem hanya dapat mendeteksi wajah dengan posisi tegak lurus.
4. Pada sistem ini menggunakan pelacakan fitur mata yang akan dianalogikan menjadi gerakan *pointer*.
5. Pada sistem ini fungsi klik kanan menggunakan kedipan mata kanan, fungsi klik kiri menggunakan kedipan mata kiri dan fungsi *double* klik menggunakan kedipan mata kiri yang lebih lama.
6. Menggunakan algoritma Lucas-Kanade untuk melakukan pelacakan pada mata.
7. Sistem ini dibangun menggunakan Microsoft Visual Studio 2012 dengan bahasa pemrograman C#.

2. Metode

2.1. Perancangan Sistem

Di dalam sistem hanya terdapat satu aktor yaitu pengguna. Pengguna ini yang nantinya akan

menggunakan sistem yang telah terpasang atau ditanamkan dalam sistem operasi komputer yang digunakan. Tugas pengguna sebagai aktor dapat terlihat pada diagram *use case* pada Gambar 1.

*

Gambar 1. Diagram *use case* sistem

Diagram aktivitas dari pilih *webcam* ditunjukkan oleh Gambar 2. Aktivitas dimulai dengan mengkoneksikan *webcam* yang akan digunakan oleh pengguna. Setelah itu sistem melakukan pengecekan koneksi untuk mengetahui *webcam* apa saja yang telah terkoneksi dengan komputer. Apabila terdapat *webcam* yang terkoneksi maka sistem akan mendapatkan properti dari *webcam* tersebut untuk dapat ditampilkan sehingga pengguna dapat memilih *webcam* mana yang akan digunakan. Pengguna juga dapat mengganti *webcam* yang digunakan ketika sistem sudah dijalankan.

Gambar 2. Diagram aktivitas pilih *webcam*

Diagram aktivitas dari jalankan sistem ditunjukkan oleh Gambar 3. Aktivitas dimulai oleh pengguna dengan melakukan klik atau menekan tombol **start**. Tombol **start** yang tertekan akan mengaktifkan *webcam* dan semua fitur pada sistem dapat berfungsi. Tombol **start** yang telah tertekan tersebut selanjutnya akan berperan sebagai tombol **stop**. Tombol **stop** ini digunakan untuk menghentikan semua proses yang sedang berjalan.

Gambar 3. Diagram aktivitas jalankan sistem

Diagram aktivitas dari deteksi wajah dan mata ditunjukkan oleh Gambar 4. Aktivitas dimulai oleh pengguna dengan meletakkan wajahnya berada di depan *webcam* yang digunakan dalam sistem. *Webcam* akan menangkap citra wajah dari pengguna tersebut lalu sistem pertama-tama akan melakukan pra-pengolahan pada citra wajah tersebut. Pra-pengolahan dilakukan dengan melakukan pencerminan, pengubahan ukuran menjadi 320x240 piksel, konversi citra BGR menjadi citra aras keabuan, dan melakukan ekualisasi histogram pada citra. Lalu pada citra hasil pra-pengolahan tersebut dilakukan proses pendeteksian wajah dengan metode Viola-Jones dan dicari wajah yang memiliki ukuran yang terbesar, jika tidak terdapat citra wajah maka sistem tidak akan melakukan aktivitas selanjutnya. Citra wajah yang terdeteksi akan ditampilkan dan akan dilakukan penentuan area mata berdasarkan pada geometri wajah. Hasil dari proses tersebut adalah berupa area mata kanan dan kiri yang ditampilkan.

Gambar 4. Diagram aktivitas deteksi wajah dan mata

Diagram aktivitas dari pelacakan mata ditunjukkan oleh Gambar 5. Aktivitas diawali oleh pengguna dengan mengubah posisi mata pada *frame* yang dilakukan dengan menggeleng-gelengkan kepala. Lalu *webcam* akan menangkap citra. Sistem akan melakukan pengecekan apakah fitur di area mata telah terdeteksi atau belum. Apabila fitur belum terdeteksi, maka sistem akan melakukan pencarian *good feature* pada area mata. Apabila fitur pada area mata telah terdeteksi maka sistem akan melanjutkan ke tahap yang berikutnya yaitu melakukan penjejakan terhadap fitur yang telah terdeteksi menggunakan algoritma Lucas-Kanade sehingga didapatkan posisi fitur saat ini. Besarnya pergerakan dapat dihitung dengan mengurangkan koordinat x dan y posisi fitur yang sebelumnya dengan koordinat x dan y posisi fitur yang sekarang lalu dikalikan dengan skala

pergerakan. Hasil kali dengan skala pergerakan tadi tambahkan dengan koordinat x dan y posisi *pointer* saat ini untuk mendapatkan koordinat x dan y posisi *pointer* berikutnya. Lalu *pointer* akan bergerak menuju posisi koordinat yang telah kita hitung tadi. Setelah itu nilai posisi fitur yang sebelumnya akan diganti dengan posisi fitur yang sekarang. Sistem melakukan perulangan proses-proses tersebut sampai sistem dihentikan atau sampai wajah pengguna tidak terdeteksi kembali.

o

menjadi 255 sedangkan apabila nilai piksel lebih dari nilai *threshold* maka nilai piksel tersebut menjadi 0. Operasi *closing* dilakukan untuk menutup lubang-lubang pada citra bola mata yang dikarenakan pantulan cahaya dari layar atau lampu led *webcam* sehingga ketika dilakukan *thresholding* area yang terkena pantulan tersebut akan menjadi hitam dan membuat lubang pada citra bola mata. Penentuan dilakukan dengan menentukan jumlah titik putih yang terdapat pada area mata, apabila jumlah piksel putih lebih kecil dari nilai batas maka dipastikan bahwa pengguna sedang berkedip sedangkan bila jumlah piksel putih lebih besar dari nilai batas maka dipastikan bahwa pengguna sedang melihat. Kesalahan-kesalahan pada pendeteksian seperti mengedipkan kedua mata secara bersamaan dan berkedip terlalu cepat dapat dapat diminimalisir dengan melakukan pengecekan apakah pada kedua mata terdeteksi sebuah kedipan atau tidak, lalu menghitung banyaknya *frame* yang terdeteksi sebuah kedipan, jika jumlahnya sudah mencapai 10 *frame* maka akan dilakukan fungsi klik kanan atau kiri dan apabila sudah mencapai 20 *frame* maka akan dilakukan fungsi *double* klik pada posisi *pointer* sesuai dengan mata yang terdeteksi dan jika kurang maka tidak dilakukan fungsi klik kanan atau kiri tersebut, untuk fungsi *double* klik hanya menggunakan mata kiri saja.

4

Gambar 5. Diagram aktivitas pelacakan mata

Diagram aktivitas dari deteksi kedipan ditunjukkan oleh Gambar 6. Aktivitas diawali oleh pengguna dengan mengedipkan mata kanan atau mata kiri. Lalu *webcam* akan menangkap citra. Setelah itu pendeteksian kedipan mata akan dilakukan, pertama-tama sistem akan meningkatkan *contrast* dari citra untuk mempertinggi perbedaan detail warna pada citra sehingga perbedaan daerah kulit dengan mata semakin jelas, dengan landasan bahwa warna bola mata cenderung mempunyai warna yang gelap dibandingkan bagian wajah lainnya (selain rambut), ketika mata terbuka di daerah mata masih terdapat piksel hitam yang merupakan bagian bola mata, sedangkan ketika mata tertutup daerah mata hampir tidak mempunyai nilai piksel hitam. Lalu dilakukan penghalusan citra menggunakan tapis gaussian untuk menghilangkan *noise-noise* pada citra. *Thresholding* citra menjadi citra biner dilakukan untuk mempermudah penghitungan piksel pada bola mata, apabila nilai piksel kurang dari nilai *threshold* maka nilai piksel tersebut akan

Gambar 6. Diagram aktivitas deteksi kedipan

3. Hasil dan Analisis

3.1 Pengujian Gerakan Pointer

Pengujian dilakukan dengan mengukur waktu yang dibutuhkan untuk menjangkau empat titik sudut layar dari posisi awal di tengah layar sampai kembali lagi ke titik awal sebanyak 30 kali dari 3 pengguna dengan tingkat pergerakan tinggi (konstanta pembagi pada sumbu $x = 15$ dan $y = 10$), normal (konstanta pembagi pada sumbu $x = 25$ dan $y = 20$) dan rendah (konstanta pembagi pada sumbu $x = 35$ dan $y = 30$) yang dilakukan di ruangan dengan kondisi penerangan baik. Pengujian ini bertujuan untuk mengukur kecepatan perpindahan *pointer*.

Berikut merupakan hasil perbandingan lama waktu pada ketiga skala gerak yang ditampilkan pada Tabel 1.

Tabel 1. Hasil rata-rata waktu yang digunakan dari ketiga tingkat pergerakan

Tingkat pergerakan	Rata-rata waktu (dalam detik)
Tinggi	5,9
Normal	7,9
Rendah	9,7

Berdasarkan data hasil pengujian pada Tabel 1, terlihat bahwa semakin besar tingkat skala gerak maka akan semakin sedikit pula waktu yang dibutuhkan untuk menggerakkan *pointer* dari suatu titik ke titik yang lain dan semakin kecil tingkat skala gerak maka akan semakin besar pula waktu yang dibutuhkan.

3.2 Pengujian Fungsi Klik

Pengujian dilakukan untuk mengukur tingkat keberhasilan melakukan klik dengan menggunakan keypad kiri dan kanan. Pengujian dilakukan sebanyak 30 kali dari 3 pengguna untuk masing-masing fungsi klik yang dilakukan di ruangan dengan kondisi penerangan baik dengan menggunakan 2 *webcam* yang berbeda.

Tabel 2. Hasil presentase dari ketiga fungsi klik

Fungsi klik	Presentase keberhasilan	
	Webcam 16 MP	Webcam 0,3 MP
Klik kiri	90%	80%
Klik kanan	86,7%	80%
Double klik	86,7%	76,7%
Rata - rata	87,8%	78,9%

Berdasarkan hasil data pengujian pada Tabel 2, terlihat bahwa resolusi *webcam* yang digunakan berpengaruh terhadap tingkat keberhasilan fungsi klik yang dilakukan. Semakin kecil resolusi atau tingkat ketajaman citra yang dihasilkan oleh *webcam* digunakan maka akan semakin besar pula tingkat kegagalan yang terjadi dan semakin besar resolusi atau tingkat ketajaman citra yang dihasilkan oleh *webcam* digunakan maka akan semakin kecil pula tingkat kegagalan.

3.3 Pengujian Tingkat Keakuratan

Pengujian dilakukan dengan menghitung waktu yang dibutuhkan untuk membuka program pada 10 *shortcut* yang terdapat pada layar dengan ketiga tingkat pergerakan oleh 3 pengguna yang dilakukan di ruangan dengan kondisi penerangan baik, seperti yang terlihat pada Gambar 7. Pengujian ini bertujuan untuk mengukur tingkat akurasi terhadap pergerakan *pointer* dan ketepatan dalam melakukan klik.



Gambar 7. Tampilan shortcut yang dijadikan sebagai pengujian

Tabel 3. Perbandingan hasil pada ketiga tingkat gerak

Tingkat gerak	Presentase	Rata-rata waktu (dalam detik)
Tinggi	77,7	7,5
Normal	86,7%	5,9
Rendah	90%	8,2

Berdasarkan data hasil percobaan pada Tabel 3, terlihat bahwa semakin tinggi skala gerak akan semakin kecil tingkat akurasi yang dihasilkan dan rata-rata waktu terkecil terdapat pada tingkat skala gerak normal. Terlalu besar skala gerak yang digunakan akan mengakibatkan *pointer* lebih sulit untuk dikontrol dan jika terlalu kecil skala gerak yang digunakan maka akan mengakibatkan terlalu lama dalam melakukan sebuah pergerakan.

3.4 Faktor-Faktor Yang Mempengaruhi

Berdasarkan penelitian yang dilakukan dapat ditemukan faktor-faktor yang mempengaruhi hasil pendeteksian, keakuratan maupun pergerakan.

Tingkat kecerahan sangat mempengaruhi tingkat keberhasilan pada pendeteksian keypad. Citra yang memiliki tingkat penerangan yang buruk akan menyebabkan kurang maksimalnya citra biner yang dihasilkan, sehingga perlu diperlukan penyesuaian nilai *threshold* dan nilai *multiply*. Tingkat kecerahan ini juga

mempengaruhi proses pendeteksian wajah pengguna, semakin baik tingkat pencahayaan maka akan semakin besar tingkat keberhasilan pendeteksian wajah.

Sistem pendeteksian wajah yang dibangun hanya dapat mendeteksi wajah dengan posisi tegak lurus. Apabila wajah pengguna terlalu menoleh kanan maupun kiri, terlalu menunduk atau terlalu menengadahkan tidak akan dapat terdeteksi.

Penentuan area mata berdasarkan geometri wajah memiliki hasil yang kurang maksimal. Hal ini disebabkan karena alis dari pengguna ikut masuk pada area hasil penentuan tersebut. Jika alis mata dari pengguna terlalu tebal maka akan dapat mempengaruhi citra biner hasil proses *thresholding* yang pada akhirnya akan mengganggu proses penentuan terjadinya kedipan.

Jika pergerakan yang dihasilkan pengguna terlalu cepat maka algoritma Lucas-Kanade tidak dapat melakukan penjejakan pada fitur yang telah dideteksi dan akan dilakukan proses penjejakan pada fitur yang baru. Hal ini akan mempengaruhi tingkat stabilitas dalam pergerakan *pointer* yang dihasilkan.

4. Kesimpulan

Algoritma Lucas-Kanade dapat berjalan dengan cukup baik untuk melakukan pelacakan fitur mata.

Berdasarkan hasil pengujian kecepatan pergerakan *pointer* yang dilakukan sebanyak 30 kali percobaan pada masing-masing skala gerak, diperoleh rata-rata waktu pada skala gerak tinggi sebesar 5,9 detik, pada skala gerak normal sebesar 7,9 detik dan pada skala gerak rendah sebesar 9,7 detik.

Berdasarkan hasil pengujian tingkat keberhasilan melakukan fungsi klik yang dilakukan sebanyak 30 kali percobaan pada setiap fungsi dan dilakukan pada dua *webcam* yang beresolusi berbeda, diperoleh tingkat keberhasilan pada *webcam* 16 MP sebesar 87,8% sedangkan pada *webcam* 0,3 MP sebesar 78,9%.

Berdasarkan hasil pengujian tingkat keakuratan sistem yang dilakukan sebanyak 30 kali percobaan pada setiap skala gerak, diperoleh presentase tingkat keberhasilan pada skala gerak tinggi sebesar 77,7% dengan rata-rata waktu sebesar 7,5 detik, presentase tingkat keberhasilan pada skala gerak normal sebesar 86,7% dengan rata-rata waktu sebesar 5,9 detik dan presentase tingkat keberhasilan pada skala gerak rendah sebesar 90% dengan rata-rata waktu sebesar 8,2 detik.

Terdapat beberapa faktor yang mempengaruhi tingkat keberhasilan sistem yaitu tingkat kecerahan, orientasi wajah pengguna, penentuan area mata yang kurang maksimal dan kecepatan pergerakan pengguna.

Referensi

- [1]. Ahmad, U., 2005, *Pengolahan Citra Digital & Teknik Pemrogramannya*, Graha Ilmu, Yogyakarta.
- [2]. Bijaksana, D., E. Setiawan dan J. Aditya, 2011, *Analisis Dan Perancangan Aplikasi Pendeteksian Kondisi Mata Tertutup Menggunakan Pustaka OpenCV*, Jurusan Teknik Informatika Binus, Jakarta.
- [3]. Bradski, G. and A. Kaehler, 2008, *Learning OpenCV*, O'Reilly Media, California.
- [4]. Fadilisyah, 2007, *Computer Vision dan Pengolahan Citra*, Andi, Yogyakarta.
- [5]. Fairhurst, M., 1998, *Computer Vision for Robotic Systems an Introduction*, Prentice Hall, New Jersey.
- [6]. Lucas, B., and T. Kanade, 1981, *An Iterative Image Registration Technique with an Application to Stereo Vision*, Carnegie-Mellon University, Pennsylvania.
- [7]. Mai, L., 2010, *Introduction to Image Processing and Computer Vision*, Institute of Information Technology, Hanoi.
- [8]. Nixon, M., and A. Aguado, 2002, *Feature Extraction & Image Processing*, Oxford, Newnes.
- [9]. Pramana, A., 2011, *Perangkat Lunak Untuk Mengendalikan Pointer Dengan Mata Menggunakan Metode Haar Cascade Dan Eye Tracking*, Jurusan Teknik Informatika UNIKOM, Bandung.
- [10]. Putra, D., 2010, *Pengolahan Citra Digital*, Andi, Yogyakarta.
- [11]. Rizzal, M., 2012, *Aplikasi Pengendali Pointer Mouse Dengan Wajah Dan Indera Penglihatan*, Jurusan Teknik Informatika UNIKOM, Bandung.
- [12]. Rodriguez, I., 2010, *Eye Tracking for Adaptive Computer Mediated Visual Interaction*, University of London.
- [13]. Trisnadik, N., 2013, *Pendeteksian Posisi Plat Nomor Kendaraan Menggunakan Metode Morfologi Matematika*, Jurusan Teknik Elektro UNDIP, Semarang.
- [14]. Victor, A. Stanley dan J. Klara, 2009, *Studi Awal Pengembangan Piranti Penunjuk Berbasis Gerakan Mata Menggunakan Pustaka OpenCV*, Jurusan Teknik Informatika Binus, Jakarta.
- [15]. Viola, P., and M. Jones, 2004, *Robust Real-Time Face Detection*, International Journal of Computer Vision, Netherlands.
- [16]. Wildana, F., 2013, *Sinkronisasi File Pada Suatu Cluster Menggunakan Aplets*, Jurusan Teknik Elektro UNDIP, Semarang.
- [17]. --, *Brightness*, <http://en.wikipedia.org/wiki/Brightness>, diakses pada 20 Februari 2013.
- [18]. --, *C sharp*, http://id.wikipedia.org/wiki/C_sharp, diakses pada 20 Februari 2013.
- [19]. --, *Histogram equalization*, http://en.wikipedia.org/wiki/Histogram_equalization, diakses pada 20 Februari 2013.
- [20]. --, *Mata*, <http://id.wikipedia.org/wiki/Mata>, diakses pada 20 Februari 2013.
- [21]. --, *Pointer (graphical user interfaces)*, [http://en.wikipedia.org/wiki/Pointer_\(graphical_user_interfaces\)](http://en.wikipedia.org/wiki/Pointer_(graphical_user_interfaces)), diakses pada 20 Februari 2013.
- [22]. --, *Unified Modeling Language*, http://id.wikipedia.org/wiki/Unified_Modeling_Language, diakses pada 20 Februari 2013.
- [23]. --, *Kamera web*, http://id.wikipedia.org/wiki/Kamera_web, diakses pada 20 Februari 2013.