

PEMBUATAN APLIKASI *COMPUTER VISION* UNTUK DETEKSI GERAKAN MENGGUNAKAN *WEBCAM* BERDASARKAN CITRA LATAR DALAM RUANGAN TERTUTUP

Youllia Indrawaty N., M. Ichwan, Lingga Satriya Yudha
Jurusan Teknik Informatik Institut Teknologi Nasional Bandung
Jl. P.K.H. Mustafa No. 23, Bandung 40124
Telp : 022-7272215 Fax : 022-7202892
youllia@itenas.ac.id, ichwan@itenas.ac.id

Abstract

Computer vision is a kind of image processing which is related with application that using visual aspect process by machine, in this case, computer. For example, many people use surveillance camera in these day. For security issue, surveillance camera takes important role for handling criminal situation in an area where there are view people. However, the monotony of surveillance camera that needs direct watch by human, and general motion detection surveillance camera that actually 'less sensitive' in detecting movement become problems in use of this kind of technology. Therefore, innovation is needed to erase all anxiety as told before. One of the method for perfecting video record is using movement detection by Frame Difference method that comparing every pixel in an on going frame with pixel with same position in previous frame. Thus, every movement or other objects beside background image can be detected sooner.

Keyword: Computer vision, surveillance camera, frame difference

1. Pendahuluan

1.1. Latar Belakang

Segala bentuk tindak kriminal yang merajalela membuat manusia semakin waspada akan dampak dari tindakan tersebut. Maka dari itu manusia selalu memperbaharui kinerja sistem keamanan dengan bantuan mesin. Segala ancaman kriminal pun dapat segera ditindaklanjuti ataupun diantisipasi. Teknologi yang saat ini dikembangkan yaitu *computer vision*. Intisari dari *computer vision* itu sendiri adalah suatu pengolahan citra yang lebih banyak berkaitan dengan aplikasi yang memanfaatkan aspek visual yang diolah oleh mesin, dalam hal ini komputer. Banyak penelitian yang telah dilakukan di berbagai negara untuk mengembangkan aplikasi *computer vision* agar dapat langsung dimanfaatkan di dalam kehidupan masyarakat. Salah satu contoh nyata yang telah menerapkan *computer vision* yaitu *surveillance camera* (kamera pemantau). *Surveillance camera* yang sering digunakan berupa perekaman keadaan suatu ruangan (*indoor* atau *outdoor*). Masalah yang dihadapi yaitu kamera hanya dapat merekam keseluruhan obyek secara monoton. Hal ini menjadi tidak efektif apabila terlalu banyak citra yang akan diamati, dan hal ini memungkinkan terjadinya kesalahan. Selain itu, seringkali pada aplikasi sistem pemantauan ini perekaman obyek dilakukan secara terus menerus tanpa kehadiran operator di muka layar komputer. Citra yang ditangkap oleh kamera disimpan dalam file berformat gambar atau video sehingga dapat diperiksa berulang-ulang kapan saja. Penyimpanan video secara terus menerus tentu saja akan memerlukan tempat penyimpanan berkapasitas sangat besar, sedangkan pada kebanyakan waktu video tersebut tidak berisi informasi penting. Oleh karena itu perlu dibuat sistem yang dapat menghemat penyimpanan video namun masih tetap berfungsi dengan baik untuk keperluan pemantauan atau keamanan. Sistem yang akan dibuat hanya merekam video apabila terdapat gerakan yang tertangkap oleh *webcam* sesuai dengan citra latar ruangan yang dipakai. Untuk dapat merealisasikan sistem tersebut, maka digunakanlah metode *frame difference* dengan membandingkan setiap piksel dalam sebuah *frame* yang sedang berlangsung dengan piksel pada posisi yang sama dalam *frame* sebelumnya. Dengan demikian, gerakan atau obyek asing yang masuk dalam suatu ruangan dapat segera terdeteksi.

1.2. Tujuan

Adapun tujuan yang hendak dicapai dalam penelitian ini adalah membuat sebuah aplikasi *computer vision* yang dapat mendeteksi gerakan suatu obyek asing di dalam ruangan yang jarang terjadi berbagai macam pergerakan (rumah / kamar yang sedang ditinggalkan, toko yang sudah tutup, isi brankas bank, dan lain-lain). Diharapkan dengan penelitian ini dapat segera diimplementasikan didalam suatu bangunan yang contohnya telah disebutkan sebelumnya sehingga keamanan gedung tersebut dapat lebih terkendali.

1.3. Rumusan Masalah

Dari penjelasan latar belakang di atas, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana kriteria ruangan tertutup yang dapat dipantau oleh *webcam*
2. Obyek seperti apa yang dapat dikenal oleh *webcam*
3. Berapa jarak obyek yang dapat ditangkap oleh *webcam*

2. Landasan Teori

2.1. Citra ^[2]

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

2.2. Pengolahan Citra ^[2]

Agar citra yang mengalami gangguan mudah diinterpretasi (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Bidang studi yang menyangkut hal ini adalah **pengolahan citra** (*image processing*). Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, sehingga menjadi citra yang kualitasnya lebih baik. Sebagai contoh, citra burung nuri pada Gambar 1 (a) tampak agak gelap, lalu dengan operasi pengolahan citra kontrasnya diperbaiki sehingga menjadi lebih terang dan tajam (b).



Gambar 1 (a) Citra burung nuri yang agak gelap,
(b) Citra burung yang telah diperbaiki kontrasnya

2.3. Frame Difference ^[3]

Frame differencing adalah suatu teknik dimana komputer memeriksa perbedaan diantara dua *frame* video. Jika terdapat perubahan pada piksel berarti terdapat sesuatu yang berubah di dalam citra (contohnya pergerakan). Kebanyakan teknik bekerja dengan blur dan *threshold*, untuk menjauhkan gerakan asli dari *noise* (gangguan). Karena frame dapat berubah juga ketika kondisi pencahayaan di dalam ruangan berubah (juga *auto focus* pada kamera, perbaikan *brightness*, dan lain-lain). ^[4] Kekurangan utama pada metode ini adalah bahwa untuk benda dengan nilai intensitas merata, pixel diinterpretasikan sebagai bagian dari latar belakang. Masalah lain adalah obyek harus terus bergerak. Jika suatu benda diam di tempat selama lebih dari periode sebuah *frame* (1/fps), maka obyek tersebut akan dianggap bagian dari latar belakang. Metode ini memiliki dua keuntungan. Salah satu keuntungan yang jelas terlihat adalah beban komputasi yang sederhana. Keuntungan lainnya yaitu model latar belakangnya yang sangat adaptif. Karena latar belakang berdasarkan pada frame sebelumnya, metodenya dapat beradaptasi dengan perubahan dalam latar belakang lebih cepat dari metode lainnya (sekitar 1/fps). Metode *frame difference* mengurangi *noise* pada latar belakang (seperti pohon yang bergoyang) jauh lebih baik dari perkiraan median yang lebih kompleks dan metode Gaussians campuran. Tantangan dalam menggunakan metode ini adalah menentukan nilai ambang batas.

2.4. Webcam ^[6]

Kamera web atau kamera ramatraya (bahasa Inggris: *webcam*, singkatan dari *web* dan *camera*) adalah sebutan bagi kamera waktu-nyata (bermakna keadaan pada saat ini juga) yang gambarnya bisa dilihat melalui Waring Wera Wanua, program pengolah pesan cepat, atau aplikasi pemanggilan video. Istilah kamera ramatraya merujuk pada teknologi secara umumnya, sehingga kata ramatraya kadang-kadang diganti dengan kata lain yang memerikan pemandangan yang ditampilkan di kamera, misalnya *StreetCam* yang memperlihatkan pemandangan jalan. Ada juga *Metrocam* yang memperlihatkan pemandangan panorama kota dan perdesaan, *TraffiCam* yang digunakan untuk memantau keadaan jalan raya, cuaca dengan *Weather Cam*, bahkan keadaan gunung berapi dengan *VolcanoCam*. Kamera ramatraya adalah sebuah kamera video bergana (*digital*) kecil yang dihubungkan ke komputer melalui (biasanya) konektor USB atau pun konektor COM.

2.5 Model Warna RGB ^[5]

Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu Red, Green dan Blue. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Apabila kita menambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255, 0, 0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Demikian apabila cahaya kita ganti dengan hijau atau biru.

3. Metode Penelitian

Perbandingan antara citra statis dengan citra yang sedang direkam menggunakan metode *frame difference* dengan rumus perhitungan yang dijabarkan sebagai berikut:

$$\begin{aligned} \text{SelisihR} &= \text{Abs}(R - R_l) \\ \text{SelisihG} &= \text{Abs}(G - G_l) \\ \text{SelisihB} &= \text{Abs}(B - B_l) \end{aligned} \quad \dots\dots\dots(1)$$

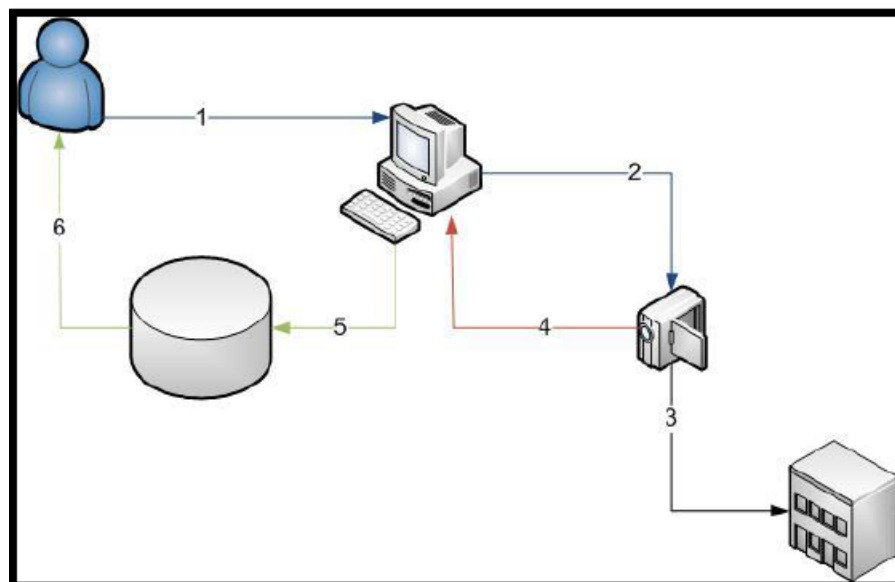
$$\begin{aligned} \text{Cacah} &= 0 \\ \text{Untuk semua piksel, jika } (\text{selisihR} > \text{ambang}) \text{ dan } (\text{selisihG} > \text{ambang}) \text{ dan } (\text{selisihB} > \text{ambang}) \text{ maka } \text{cacah} &+ 1 \\ \text{Jika } (\text{cacah} > \text{persentase}) \text{ maka gerakan terdeteksi} \end{aligned} \quad \dots\dots\dots(2)$$

Pada persamaan 1 perlu dilakukan pemisahan ketiga elemen warna (R = red, G = green, Blue = blue) dalam menghitung selisih absolut. Caranya yaitu dengan mengalikan nilai absolut terhadap pengurangan nilai warna citra asal dengan warna citra latar. Obyek bergerak yang akan dideteksi biasanya memiliki ukuran yang cukup besar, oleh karena itu perlu ditentukan banyaknya piksel yang bergerak dalam setiap *frame*. Apabila jumlah piksel bergerak relatif kecil, maka ada kemungkinan obyek tersebut adalah bagian dari latar. Dengan demikian perlu dihitung banyaknya piksel yang selisihnya melebihi ambang. Hal ini dapat dilakukan menggunakan algoritma pada persamaan 2, dimana apabila banyaknya piksel yang selisihnya melebihi ambang (variabel cacah dengan satuan persentase terhadap banyaknya seluruh piksel dalam *frame*) lebih dari suatu nilai persentase yang diberikan, maka dapat disimpulkan bahwa telah terdeteksi suatu gerakan.

4. Hasil Dan Pembahasan

4.1 Workflow Keseluruhan Sistem

Pada sub-bab ini dijelaskan perancangan yang proses kerjanya terdapat elemen-elemen yang saling berkaitan satu sama lain. Proses kerja dimulai dari awal pengoperasian aplikasi hingga akhirnya didapat output setelah aplikasi berjalan.



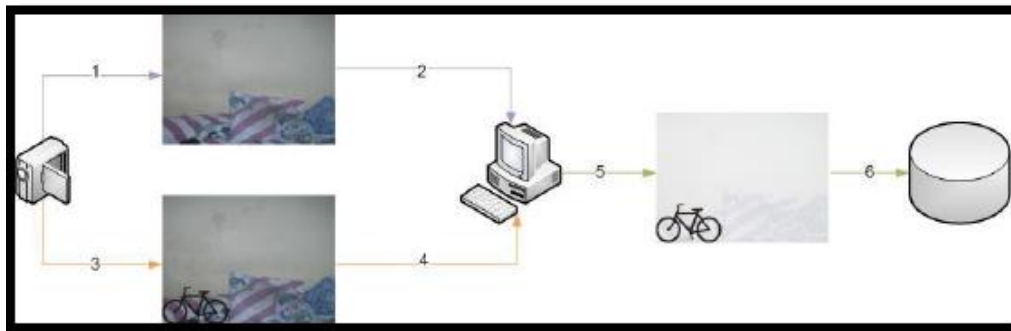
Gambar 2. Workflow Keseluruhan Sistem

Keterangan:

1. *User* mulai mengoperasikan aplikasi menggunakan komputer.
2. Aplikasi dapat digunakan apabila sebelumnya telah terpasang *webcam* pada *server*.
3. *Webcam* yang telah dipasang pada komputer *user* dapat digunakan dalam ruangan tertutup pada suatu bangunan.
4. Kemudian *webcam* akan memberikan *feedback* berupa informasi citra yang berhasil ditangkap oleh *webcam* ke dalam komputer *user*.
5. Data mentah yang dihasilkan *webcam* kemudian diolah dan dijadikan data permanen hasil perekaman *webcam*.
6. Data yang telah diolah sebelumnya dapat diperlihatkan kepada *user*.

4.2 Workflow Aplikasi

Pada sub-bab ini dijelaskan perancangan yang proses kerjanya menyangkut cara kerja aplikasinya saja. Perancangan ini bersifat *internal* yang berarti hanya menjelaskan bagaimana aplikasi dapat mendeteksi gerakan sesuai dengan input yang diberikan oleh *user*.



Gambar 3. Workflow Aplikasi

Keterangan:

1. *Webcam* meng-*capture* latar ruangan tertutup.
2. Citra latar dijadikan data sementara, kemudian data tersebut disimpan dalam komputer.
3. *Webcam* merekam citra yang sedang berlangsung.
4. Citra yang sedang berlangsung dijadikan data sementara, kemudian data tersebut disimpan dalam komputer.
5. Komputer mengolah citra yang telah didapat oleh *webcam*, kemudian mengolah citra tersebut dengan cara membandingkan citra yang sedang berlangsung dengan citra latar yang telah di-*capture* menggunakan metode *frame difference*.
6. Setelah terdeteksi adanya obyek asing, maka citra hasil perbandingan dijadikan data permanen.

4.3 Pengujian Sistem

4.3.1 Pengujian dengan Intensitas Cahaya Terang

Pengujian sistem ini menggunakan metode pengujian *black box*. Metode *black box* digunakan untuk mengecek apakah aplikasi sudah berjalan sesuai dengan *requirement* yang diimplementasikan atau belum. Pengujian ini memungkinkan analisis sistem memperoleh kumpulan kondisi input yang akan mengerjakan seluruh kumpulan fungsional aplikasi. Berikut merupakan interface utama dari implementasi aplikasi yang dibangun.

Kondisi yang diberikan untuk menguji aplikasi deteksi gerakan ini berada di dalam ruangan tertutup dengan spesifikasi pengujian sebagai berikut:

- a) Luas ruangan berukuran 360 x 300 cm.
- b) Intensitas cahaya terang menggunakan sumber cahaya lampu dengan daya 18 Watt; intensitas cahaya gelap menggunakan sumber cahaya alami.
- c) Latar yang dipakai berupa banyak obyek dan latar polos.
- d) Obyek asing yang diuji berupa “tangan”, “pulpen”, “botol kecil”, dan “sebatang kayu”.
- e) Nilai ambang yang diberikan adalah 9 dan nilai persentasenya adalah 1% untuk intensitas cahaya terang; sedangkan nilai ambang yang diberikan adalah 6 dan nilai persentasenya adalah 1% untuk intensitas cahaya gelap. Nilai-nilai tersebut merupakan nilai dengan tingkat sensitivitas tertinggi



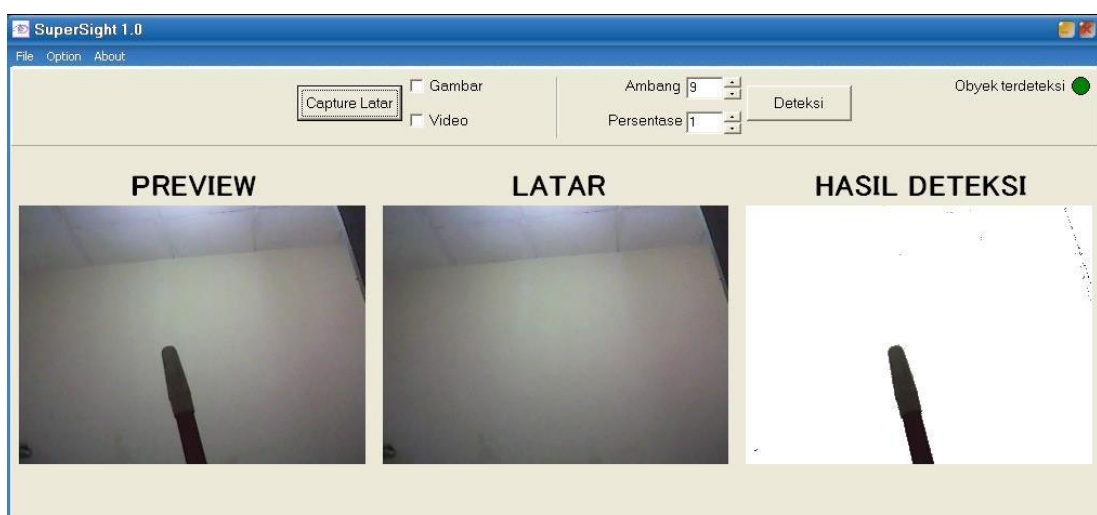
Gambar 4 Interface aplikasi (obyek asing tidak terdeteksi)

Pada Gambar 4, *user* diminta untuk mengcapture latar statis seperti yang terlihat pada tampilan “preview”. Citra latar ruangan akan tampak pada tampilan “latar”. Pada tampilan “hasil deteksi” terlihat putih saja karena tidak ada pergerakan atau obyek asing yang masuk kedalam ruangan.



Gambar 5 Interface aplikasi (obyek asing terdeteksi)

Pada Gambar 5, terlihat adanya pergerakan atau obyek asing, yaitu tangan, masuk ke dalam ruangan yang ditampilkan pada “hasil deteksi”. Hal tersebut terjadi karena sebelumnya citra latar pada “latar” dibandingkan dengan citra *real-time* pada tampilan “preview”. Perbandingan tersebut menggunakan metode *frame difference*. *Feedback* yang diberikan yaitu berupa file video keseluruhan citra dalam ruangan yang otomatis tersimpan ke dalam komputer *user* apabila obyek asing telah terdeteksi. Selain itu, sebagai pembanding, file gambar berupa obyek asing tadi juga diikutsertakan. Pada Gambar 6 di bawah ini terlihat hasil deteksi pada opsi ambang 9 dan persentase 1% untuk Intensitas Cahaya Terang.



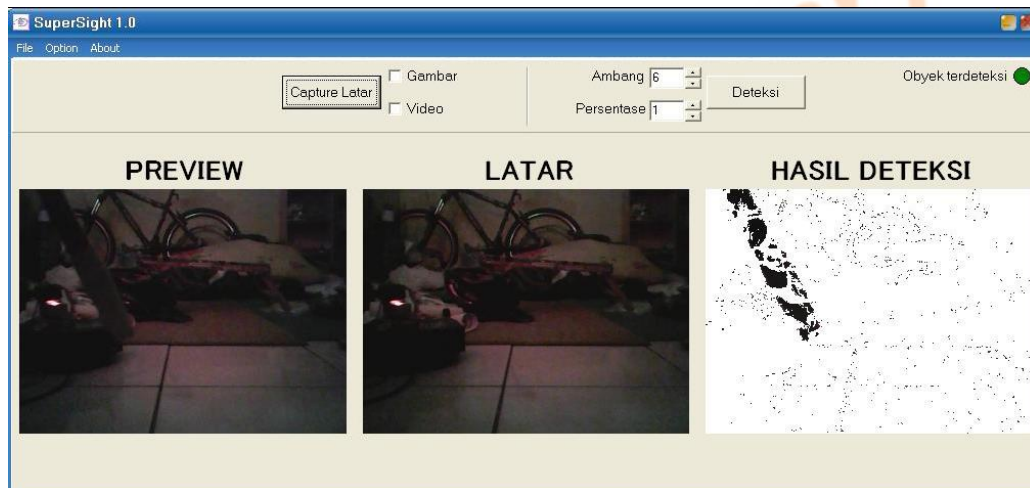
Gambar 6 Intensitas Cahaya Terang dengan obyek asing

4.3.2 Pengujian dengan Intensitas Cahaya Gelap

Pada Gambar 7 dan gambar 8 ditampilkan studi kasus dengan intensitas cahaya gelap menggunakan sumber cahaya alami yang berada di tengah ruangan. Kemudian untuk mendapatkan hasil pengujian, ditempatkan beberapa obyek asing dengan jarak yang berbeda-beda dari *webcam*. Apabila status pendeteksian menghasilkan “Obyek tidak terdeteksi”, maka obyek asing akan diganti dengan ukuran yang lebih besar hingga menghasilkan status pendeteksian “Obyek terdeteksi”.



Gambar 7 Intensitas Cahaya Gelap dengan obyek asing berukuran kecil



Gambar 8 Intensitas Cahaya Gelap dengan obyek asing berukuran besar

5. Kesimpulan

Berdasarkan proses implementasi dan pengujian yang telah dipaparkan, maka dapat disimpulkan bahwa aplikasi dapat berjalan dengan *requirement* yang dipakai. Kemampuan aplikasi untuk mendeteksi gerakan dan obyek asing masih tergantung dengan seberapa tingginya spesifikasi *webcam*, intensitas dan arah cahaya, serta warna latar ruangan yang digunakan. Maka dari itu, *user* perlu mengatur sensitivitas citra menggunakan opsi “ambang” dan “persentase”.

DAFTAR PUSTAKA

- [1] Munir, R., “Pengolahan Citra Digital dengan Pendekatan Algoritmik”, Informatika, Bandung, 2004.
- [2] Rinaldi, “Pengantar Pengolahan Citra”, [http://www.informatika.org/~rinaldi/Buku/Pengolahan Citra Digital/Bab 1_Pengantar Pengolahan Citra.pdf](http://www.informatika.org/~rinaldi/Buku/Pengolahan%20Citra%20Digital/Bab%201_Pengantar%20Pengolahan%20Citra.pdf), November 2004.
- [3] Kamperman, Kasper, “Computer Vision: Frame Differencing”, <http://www.kasperkamperman.com/blog/computer-vision/computervision-framedifferencing/>, April 2010.
- [4] Saharkiz, Aresh, “Low complexity background subtraction using frame difference method”, <http://areshmatlab.blogspot.com/2010/05/low-complexity-background-subtraction.html>, Mei 2010.
- [5] Pratomo, Herman, “Model Warna RGB”, pengantar- <http://warna.blogspot.com/2008/10/model-warna-rgb.html>, Oktober 2008.
- [6] Wikipedia, “Kamera Web”, http://id.wikipedia.org/wiki/Kamera_web, Januari 2012.