

Real Time Object Tracking Dengan Algoritma Speeded Up Robust Features Melalui WebCam

David

Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jalan Merdeka No. 372 Pontianak, Kalimantan Barat
e-mail: DavidLiau@gmail.com

Abstrak

Proses pengenalan objek pada umumnya diawali dengan pendeteksian fitur menggunakan sistem pengolahan citra untuk menangkap informasi objek yang akan dikenali. Penelitian ini bertujuan untuk mendeskripsikan cara mendeteksi objek visual dengan kecepatan tinggi secara real time melalui webcam. Dalam penelitian ini, dikembangkan aplikasi untuk mendeteksi objek dengan menggunakan algoritma speeded up robust features (SURF). Algoritma Surf yang memiliki fungsi sebagai pengenalan objek atau rekonstruksi 3D dimana algoritma ini memiliki perhitungan yang cepat dari operator dalam kotak-ruang, memungkinkan aplikasi real-time dalam pencocokan dan pengenalan objek. Tahapan yang dilakukan untuk mendapatkan deskriptor dimulai dari scanning citra, pre-processing, ekstraksi ciri dengan menggunakan SURF sampai pada proses matching pada saat pengujian. Pendeteksian Obyek Melalui Webcam Menggunakan Algoritma Speeded Up Robust Features (SURF) mampu menangkap setiap jenis Obyek yang ingin di-track untuk memudahkan pengguna dalam mencari Obyek yang sesuai dengan kriteria dari gambar Obyek yang ditangkap oleh kamera dengan gambar yang telah ditangkap dan disimpan sebagai bahan tracking. Dari hasil pengujian, dapat dilihat pada tahap pengujian telah berhasil melakukan ekstraksi citra obyek menggunakan SURF. Selain itu, dapat ditampilkan interest point pada citra yang ditandai dengan maching lines. Hasil pengujian telah berhasil membuktikan kehandalan SURF, hal ini dilihat dari pendeteksian obyek bekerja stabil pada intensitas cahaya yang cukup terang, semakin dekat jarak obyek maka obyek tersebut akan terdeteksi, masih mampu mendeteksi obyek yang dirotasi serta mampu mendeteksi obyek yang sudut miringnya tidak terlalu besar.

Kata kunci: Algoritma SURF, Blob Detection, Feature Extraction, Object Tracking, Interest Point Detection

1. Pendahuluan

Algoritma SURF merupakan algoritma yang dapat mendeteksi fitur-fitur pada obyek bergerak dengan menggunakan kecepatan komputasi box filter dengan menggunakan citra integral. Adapun hasil deteksi obyek dan deskripsi fitur-fitur lokal dari obyek dapat dipergunakan dalam tracking objek bergerak. Kelebihan

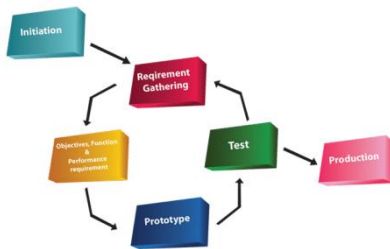
dari Algoritma SURF yaitu kecepatan dalam mendeteksi fitur-fitur obyek pada citra yang lebih cepat dibandingkan dengan metode lainnya dan memiliki ketahanan (robustness) terhadap transformasi rotasi (rotation), perubahan skala, pencahayaan, dan perubahan sudut pandang [1].

Algoritma SURF bertujuan untuk mendeteksi fitur lokal suatu citra dengan handal dan cepat [2]. Algoritma ini sebagian terinspirasi oleh algoritma SIFT (*Scale-invariant feature transform*), terutama pada tahap *scale space representation* [3]. Algoritma SURF menggunakan penggabungan algoritma citra integral (*integral image*) dan *blob detection* berdasarkan determinan dari matriks Hessian. *Speeded-Up Robust Features* (SURF) adalah sebuah algoritma ekstraksi ciri yang dapat digunakan mengatasi kelemahan dari algoritma *Linear Binary Pattern* dan *Graylevel Co-occurrence Matrix*.

Sebelumnya telah banyak yang melakukan penelitian mengenai kasus pengambilan keputusan menggunakan metode Speeded Up Robust Features (SURF) diantaranya yaitu penelitian pendeteksian rambu lalu lintas dengan algoritma *Speeded Up Robust Features* (SURF) [4]. Penelitian ini merancang sebuah sistem aplikasi yang berfungsi menggunakan algoritma *Speeded Up Robust Features* (SURF) untuk mendeteksi rambu lalu lintas, yang dapat digunakan pada kendaraan yang bergerak, dengan menggunakan algoritma speeded up robust features (SURF) [4]. Penelitian selanjutnya tentang ekstraksi *keyframe* dengan *Entropy Differences* untuk temu kembali konten video berbasis *Speeded-Up Robust Features* [5]. Penelitian ini merancang sebuah metode baru untuk CBVR berbasis SURF dengan mengadaptasi logika metode *Entropy Differences* untuk mengekstraksi *keyframe* secara efektif [5]. Penelitian berikutnya tentang analisis dan implementasi *Speeded-Up Robust Features* pada *Content Based Image Retrieval* [6]. Penelitian ini merancang sebuah sistem berbasis *content based image retrieval* yang dapat mengekstraksi ciri berdasarkan *local features* yang tahan terhadap perubahan transformasi dan dapat mengidentifikasi objek dengan lebih spesifik menggunakan algoritma SURF dikarenakan algoritma tersebut menunjukkan tingkat performansi yang terbaik dibanding yang lainnya [6].

2. Pembahasan

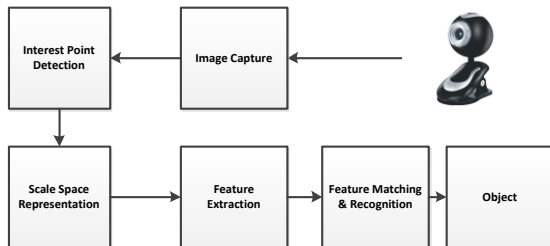
Metode perancangan perangkat lunak menggunakan sistem *prototype*. Metode *prototype* dibangun untuk mendefinisikan kebutuhan [7].



Gambar 1. Diagram langkah-langkah metode prototype

Untuk metode pengujian adalah menggunakan pengujian blackbox yaitu pengujian memungkinkan perencana perangkat lunak mendapatkan serangkaian input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pembuatan aplikasi ini menggunakan *Visual Basic .NET*.

Dalam implementasinya, algoritma SURF dibagi menjadi beberapa tahapan seperti pada gambar 2 berikut [1]:



Gambar 2. Tahapan Algoritma Surf

a. Interest Point Detection

Deteksi *interest point* digunakan untuk memilih titik yang mengandung banyak informasi dan sekaligus stabil terhadap gangguan lokal atau global dalam citra digital. Dalam algoritma SURF, dipilih detektor titik perhatian yang mempunyai sifat invarian terhadap skala, yaitu *blob detection*. Blob merupakan area pada citra digital yang memiliki sifat yang konstan atau bervariasi dalam kisaran tertentu. Blob merupakan sekelompok piksel yang terhubung dalam citra yang berbagi beberapa properti umum (misal nilai grayscale) [8]. Untuk melakukan komputasi *blob detection* ini, digunakan determinan dari matriks Hessian dari citra. Dalam algoritma SURF, determinan matriks Hessian dihitung dari wavelet Haar dengan menggunakan *integral image* nya secara optimal [8]. Determinan dari matriks Hessian digunakan sebagai dasar algoritma SURF karena sifat invarian terhadap skala, kestabilan dan berulang dengan mudah [8].

b. Scale Space Representation

Dengan ukuran citra yang berbeda-beda, akan sangat sulit bagi kita untuk membandingkan fitur-fitur yang terdapat pada citra tersebut. Maka dari itu, diperlukan

suatu proses yang menangani perbedaan ukuran dengan menggunakan metode perbandingan skala. Dalam metode ini, kita menggunakan *scale space* di mana citra diimplementasikan dalam bentuk sebuah *image pyramid* [3]. Citra secara berulang akan diperhalus (*smoothing*) dengan fungsi Gaussian dan secara beruntun dengan cara *sub-sampling* untuk mencapai tingkat tertinggi pada piramida. Dengan menggunakan *integral image*, perhitungan ini tidak perlu dilakukan secara iteratif dengan menggunakan filter yang sama, tetapi dapat filter dengan ukuran sembarang ke dalam beberapa skala citra yang berbeda.

c. Feature Description

Fitur didefinisikan sebagai bagian yang mengandung banyak informasi suatu citra, dan fitur ini digunakan sebagai titik awal untuk algoritma deteksi objek. Tujuan dari proses deteksi fitur ini adalah untuk mendapatkan deskripsi dari fitur-fitur dalam citra yang diamati. Langkah pertama adalah melihat orientasi yang dominan pada titik perhatian yang terdapat dalam citra, kemudian membangun suatu area yang akan diambil nilainya dan mencari fitur korespondensi pada citra pembanding [2]. Dalam penentuan orientasi suatu citra kita menggunakan filter wavelet Haar, disini dapat ditentukan tingkat kemiringan suatu fitur yang diamati. Selanjutnya untuk deskripsi fitur dalam algoritma SURF, digunakan hanya perhitungan gradient histogram dalam empat kelompok (bins) saja untuk mempercepat perhitungan.

d. Feature Matching and Recognition

Dalam tahap ini, dilakukan perbandingan fitur hasil perhitungan proses sebelumnya tetapi hanya bila terdapat perbedaan kontras, yang dideteksi melalui tanda dari *trace* matriks Hessian. Proses pencocokan keypoint dengan metode Fast Library Approximated Nearest Neighbor (FLANN). FLANN merupakan algoritma yang dioptimalkan untuk pencarian tetangga terdekat yang cepat dalam dataset besar dan untuk fitur dimensi tinggi. Dengan cara ini, biaya komputasi dari algoritma SURF bisa dikatakan sangat minim.

Dalam aplikasi sistem pelacakan obyek ini terdapat beberapa hardware yang dibutuhkan diantaranya kamera webcam berfungsi sebagai penangkap gambar dari objek. Laptop berfungsi sebagai pengambil data dan pengolah data yang sudah ditangkap oleh webcam. EmguCV mempunyai API (Application Programming Interface) untuk High level maupun low level, terdapat fungsi-fungsi yang siap digunakan, baik untuk loading, saving, akuisisi gambar maupun video. Program aplikasi pendeteksian obyek dibuat menggunakan perangkat lunak *Microsoft Visual Studio 2013* dengan Bahasa *Visual Basic .NET* dan pustaka *EmguCV*. Pembangunan perangkat lunak menggunakan *Visual Basic .NET* dan pustaka *EmguCV* yang terbagi atas proses preprocessing yakni tahapan awal sebelum obyek citra diolah dengan pengolahan citra digital, deteksi keypoint, pencocokan keypoint dengan metode Fast Library Approximated

Nearest Neighbor (FLANN), proses penggabungan citra, dan pemotongan citra untuk mengurangi area hitam di sekitar citra.

Preprocessing adalah proses tahap awal yang dilakukan agar obyek bisa lebih mudah diolah nantinya. Proses *Preprocessing* terdiri atas proses mengubah obyek citra yang semula dalam ruang warna RGB (Red Green Blue) diubah menjadi citra gray-scale (keabu-abuan). Tujuan konversi ini agar obyek citra yang semula berada pada tiga layer berubah menjadi satu layer saja agar proses komputasinya lebih mudah. Obyek citra yang hendak dibaca oleh sistem disimpan dalam Matriks dalam bentuk matriks 2D dengan masing-masing titik mewakili satu pixel.

Tahapan berikutnya adalah pemanggilan algoritma SURF untuk mendeteksi keypoint. Sesuai dengan algoritma SURF itu sendiri yang menggunakan perhitungan dengan pendekatan matriks Hessian, maka perlu ditentukan ambang batas dari nilai determinan Hessian agar suatu titik dapat dikatakan sebagai keypoint. Semakin besar ambang batas yang digunakan maka semakin sedikit keypoint yang terdeteksi, sebaliknya semakin kecil nilai Hessian maka semakin banyak keypoint yang terdeteksi dengan ambang batas yang sebaiknya digunakan bernilai 400 [9].

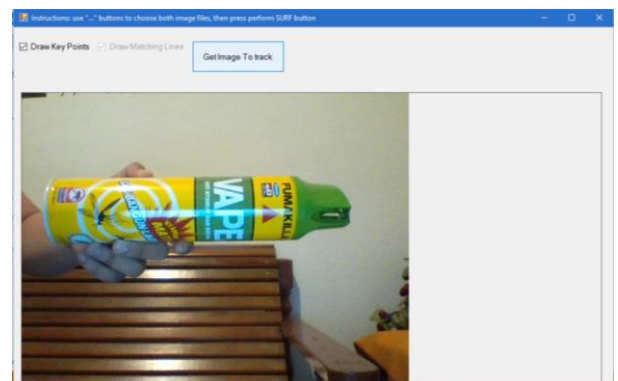
Data hasil deteksi keypoint disimpan ke dalam penampung detector. Kemudian hasil perhitungan ini dideklarasikan dalam suatu bentuk vektor, nilai keypoints dimasukkan ke dalam variabel keypoints_object untuk citra pertama dan keypoints_scene untuk citra kedua. Deskriptor merupakan daerah piksel di sekitar keypoint yang dihasilkan. Deskriptor menggambarkan distribusi intensitas piksel tetangga disekitar keypoint. Proses pertama yang dilakukan adalah mencocokkan orientasi yang dihasilkan berdasarkan informasi dari daerah yang berbentuk lingkaran disekitar piksel yang menjadi keypoint. Kemudian proses berikutnya membuat daerah berbentuk kotak pada orientasi yang terpilih dan mengekstrak deskriptor SURF dari daerah tersebut.

Implementasi dari pencocokan FLANN KNN bertujuan untuk mencocokkan masing-masing keypoint yang memiliki deskriptor pada setiap citra. Nilai hasil dari pencocokan disimpan dalam vector <Dmatch> dan hasil keluarannya disklarasikan sebagai matches. Hasil kecocokan kemudian diperiksa kembali pada setiap barisnya dengan fungsi perulangan for untuk mendapatkan nilai-nilai yang terbaik dan sesuai dengan batasan yang ditentukan. Nilai ini kemudian dideklarasikan sebagai good_matches.

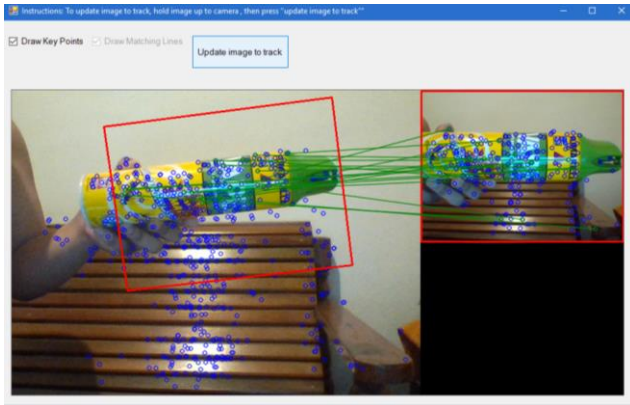
Program utama berisikan proses penggabungan yang diawali dengan langkah preprocessing dengan mengkonversi citra masukan RGB menjadi citra grayscale. Konversi citra kedua gambar dilakukan dengan memanggil sebuah fungsi dari pustaka OpenCV.

Proses selanjutnya yang merupakan proses utama adalah deteksi keypoints masing-masing citra menggunakan algoritma Speeded up Robust Features yang telah disediakan oleh pustakan OpenCV sehingga proses menjadi lebih mudah. Pada proses ini didapat keypoints yang kemudian diekstraksi untuk memperoleh deskripsi dari setiap keypoints kedua citra. Langkah selanjutnya adalah melakukan proses pencocokan keypoints menggunakan FLANN atau Fast Library Approximated Nearest Neighbor. untuk memperoleh keypoints yang saling berkesesuaian dari kedua citra masukan. Dari proses pencocokan diperoleh titik-titik kesamaan antar keypoints dari kedua citra yang membentuk suatu pola. Dari pola-pola yang terbuat tersebut kemudian dilakukan perhitungan untuk memperoleh homography matrix atau pola kesamaan dari keypoints yang berkesesuaian menggunakan algoritma RANSAC yang telah tersedia pada pustaka OpenCV. Setelah didapat pola kesamaan keypoints dilakukan warping yang merupakan proses penggabungan citra sehingga diperoleh hasil akhir berupa citra tuah berkesesuaian yang berasal dari dua masukan citra yang berbeda.

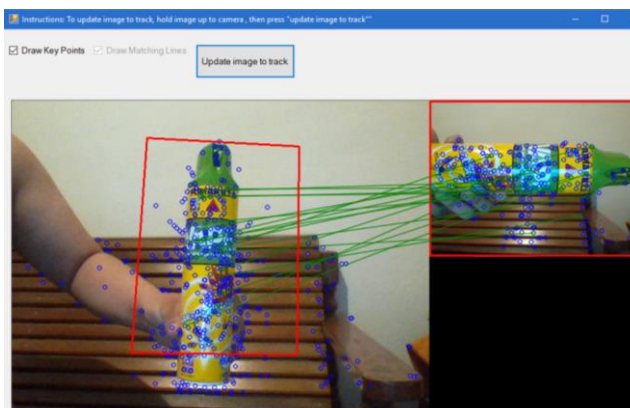
Cara Kerja program aplikasi ini adalah melakukan proses tracking obyek, diperlukan adanya cover image (gambar pembanding) yang berguna untuk proses tracking obyek. Jika Obyek tidak ditemukan, maka tracking object tidak berhasil, dan kembali kepada proses load image untuk membandingkan image yang ada pada database, dengan Load image secara real time. Mekanisme program aplikasi ini dimulai ketika pengguna melakukan masukan melalui hasil *capture* dari kamera/*webcam* kemudian melalui hasil tersebut maka dimulai proses komputasi menggunakan algoritma *integral image*. Setelah perhitungan selesai maka dilakukan *blob detection*. Langkah selanjutnya adalah program aplikasi melakukan *feature matching* dengan hasil citra model yang sebelumnya telah disimpan dalam program aplikasi. Jika terjadi kecocokan (*match*), program akan mengeluarkan hasil pengenalan dari citra tersebut. Gambar 1 berikut ini merupakan *screen shoot* dari program aplikasi yang dikembangkan.



Gambar 1. Tampilan Aplikasi



Gambar 2. Tampilan Proses Tracking Object



Gambar 3. Tampilan Proses Tracking Object dengan posisi obyek yang berbeda

Gambar 1 menggambarkan aplikasi tersebut terdapat *Image Box EmguCV* yang berguna utk menampilkan gambar yang didapat dari *webcam*/kamera. Kemudian pada gambar nomor 2 pada saat program aplikasi dieksekusi maka dengan menekan tombol ‘*Get Image To Track*’ untuk mengambil gambar dari jenis Obyek yang ingin di-*track* kemudian Obyek tersebut akan di-*capture* dan akan diletakkan di sebelah kanan dari *image box* untuk men-*track* Obyek tersebut di layar sebelahnya dari kamera/*webcam* yang tempat kita mengarahkan Obyek yang di cari/*track* sesuai atau tidak. Selanjutnya gambar 3 setelah menekan tombol ‘*Get Image To Track*’ tombol akan berubah menjadi ‘*Update Image To Track*’ yang berarti kita dapat mengubah gambar yang awal dengan gambar yang baru untuk dideteksi dan seterusnya. Dari hasil pengujian, dapat dilihat pada tahap pengujian telah berhasil melakukan ekstraksi citra obyek menggunakan SURF. Selain itu, dapat ditampilkan interest point pada citra yang ditandai dengan *matching lines*.

Pada perancangan aplikasi ini, metode pengujian yang digunakan yaitu *black-box testing*. Pengujian *black-box* untuk memastikan semua pernyataan telah diuji dan untuk mengetahui fungsi, masukan, dan keluaran dari pendeteksian obyek melalui *webcam* dengan algoritma SURF yang dirancang sesuai dengan yang dibutuhkan. Pengguna mengarahkan kamera dengan baik terlebih

dahulu kemudian menaruh posisi obyek yang akan di-*track* sesuai dengan posisi yang diinginkan kemudian menekan tombol ‘*Get Image To Track*’ maka posisi tadi akan di tangkap sebagai gambar *tracking* terhadap jenis obyek yang sesuai atau mirip dengan gambar dengan posisi yang diubah. Langkah berikutnya dengan menekan tombol *update image to track* maka setelah gambar awal sudah ingin diganti maka dengan menekan tombol tersebut pengguna dapat memperbaharui gambar yang ingin dicari selanjutnya gambar baru akan digunakan untuk *tracking*. Tahap selanjutnya adalah apabila pengguna ingin mematikan semua *tracking* yang ada maka dapat menekan tombol *draw key points* maka kotak merah dan garis-garis penghubung tidak akan muncul. Skenario terakhir adalah bila hanya ingin melihat dimana benda itu berada tanpa adanya garis-garis yang menghubungkan gambar yang ditangkap dengan gambar di kamera maka pengguna dapat menghilangkan centang pada *draw matching lines* untuk menghilangkan garis-garis tersebut. Pengujian yang telah dilakukan dimaksudkan agar setiap kesalahan yang terjadi dapat ditampilkan, sehingga pengguna yang menggunakan aplikasi Pendeteksian Obyek Melalui Webcam Dengan Algoritma SURF tahu kesalahan yang terjadi.

Untuk mengevaluasi kehandalan aplikasi diperlukan beberapa scenario pengujian lainnya yaitu melakukan pengujian terhadap intensitas cahaya seperti pada Tabel 1 berikut, pengujian terhadap jarak seperti pada tabel 2, serta pengujian terhadap sudut pandang obyek seperti pada tabel 3.

Tabel 1. Pengujian Terhadap Intensitas Cahaya

No	Intensitas Cahaya	Hasil pengujian
1	Sangat Terang (≥ 80 Lux)	Gagal
2	Terang (50 – 80 Lux)	Baik
3	Cukup (30 – 50 Lux)	Baik
4	Redup (10 – 30 Lux)	Baik
5	Sangat Redup (< 10 Lux)	Gagal

Pengujian terhadap intensitas cahaya dilakukan untuk mengetahui rentang atau batasan aplikasi deteksi obyek ini dapat bekerja dalam intensitas cahaya range tertentu. Saat kondisi cahaya sangat redup (agak gelap), aplikasi tidak dapat mendeteksi obyek atau gagal. Saat kondisi cahaya terlalu terang juga demikian aplikasi tidak dapat bekerja secara stabil terkadang berhasil terkadang gagal. Hal ini dikarenakan kondisi tingkat kecerahan cahaya tersebut mengganggu penangkapan gambar yang dilakukan oleh *web cam* sehingga menyebabkan kegagalan dalam perhitungan interest point. Aplikasi deteksi obyek bekerja stabil pada intensitas cahaya yang cukup terang.

Skenario pengujian berikutnya dengan pengujian terhadap jarak, semakin dekat jarak obyek maka obyek tersebut akan terdeteksi namun hal ini berbeda jika jarak makin jauh maka obyek tidak akan terdeteksi. Perlakuan

akan percobaan terhadap jarak ini akan mempengaruhi ukuran dan jenis objek sehingga menyebabkan kemampuan aplikasi dalam melakukan tracking sehingga diketahui batasan jarak yang baik untuk penempatan web cam (lihat hasil pada tabel 2).

Tabel 2. Pengujian Terhadap Jarak

No	Jarak	Hasil pengujian
1	Sangat Dekat (≤ 30 cm)	Gagal
2	Dekat (30 – 50 cm)	Baik
3	Cukup (50 – 100 cm)	Baik
4	Jauh (100 – 300 cm)	Baik
5	Sangat Jauh (≥ 300 cm)	Gagal

Skenario ketiga yaitu pengujian terhadap perubahan sudut pandang dan rotasi. Pada proses ini, dilakukan segala percobaan berkaitan dengan posisi obyek yang dirotasi dan diatur sedemikian sehingga sudut pandang obyek berbeda. Tampak dari hasil percobaan untuk pengujian secara rotasi bahwa aplikasi masih mampu mendeteksi obyek, namun untuk sudut pandang dengan kemiringan tertentu akan menyebabkan aplikasi tidak dapat melakukan tracking sehingga diketahui batasan ketelitian hanya untuk obyek yang sudut miringnya tidak terlalu besar.

Tabel 3. Pengujian Terhadap Sudut Pandang

No	Sudut Pandang	Hasil pengujian
1	$\geq 45^\circ$	Gagal
2	$30^\circ - 45^\circ$	Baik
3	$0^\circ - 30^\circ$	Baik

Dari hasil pengujian terhadap jarak dan tingkat pencahayaan ditemukan bahwa penentuan posisi webcam dengan obyek dan pencahayaan memiliki pengaruh besar dalam deteksi obyek, oleh karena itu dibutuhkan pengaturan web cam dan pencahayaan yang cukup. Akurasi dapat ditingkatkan dengan menambahkan ekstraksi ciri obyek berdasarkan ciri histogram warna dan bentuk yang diinginkan ke dalam algoritma tersebut, sehingga pembentukan cover image atau sebagai objek yang dideteksi dapat lebih baik walaupun dihapuskan dengan berbagai posisi sudut pandang dan rotasi obyek.

3. Kesimpulan

Pendeteksian Obyek Melalui Webcam Menggunakan Algoritma Speeded Up Robust Features (SURF) mampu menangkap setiap jenis Obyek yang ingin di-track untuk memudahkan pengguna dalam mencari Obyek yang sesuai dengan kriteria dari gambar Obyek yang di tangkap oleh kamera dengan gambar yang telah di tangkap dan disimpan sebagai bahan *tracking*. Hasil yang didapat dari sistem *tracking* dengan algoritma SURF ini sesuai atau sama dengan hasil yang dilakukan apabila dilakukan pencarian atau *tracking* secara manual yaitu menggunakan indera penglihatan. Selain itu dengan algoritma SURF hasil pencarian dapat dilakukan dengan cepat dengan menggunakan garis-garis

penghubung membuktikan bahwa banyaknya titik yang diambil lebih banyak sehingga hasil yang dihasilkan lebih teliti. Pengguna dapat mengubah jenis Obyek apa saja yang diinginkan untuk di-track. Hasil pengujian telah berhasil membuktikan kehandalan SURF, hal ini dilihat dari pendeteksian obyek bekerja stabil pada intensitas cahaya yang cukup terang, semakin dekat jarak obyek maka obyek tersebut akan terdeteksi, masih mampu mendeteksi obyek yang dirotasi serta mampu mendeteksi obyek yang sudut miringnya tidak terlalu besar. Aplikasi ini kedepannya akan diberikan tampilan antarmuka yang user-friendly dan menarik serta penggabungan dengan jaringan saraf tiruan (Neural Networks) untuk menyelesaikan aplikasi ini.

Daftar Pustaka

- [1]. J. W. Hsieh, L.C. Chen, D.Y. Chen. "Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition." *IEEE Transactions on intelligent transportation systems*, 2014, 15.1: 6-20.
- [2]. H. Bay, A. Ess, T. Tuytelaars, L. Van Gool. "Speeded-up robust features (SURF)". *Computer vision and image understanding*, 2008, 110.3: 346-359.
- [3]. D.G. Lowe. "Object recognition from local scale-invariant features". In *Computer vision*, 1999. The proceedings of the seventh IEEE international conference on. Ieee, 1999. p. 1150-1157.
- [4]. A.A. Gunawan, W. Gazali. "Pendeteksian Rambu Lalu Lintas Dengan Algoritma Speeded Up Robust Features (SURF)". *Jurnal Mat Stat*, 2013, 13.02.
- [5]. M.M. Huda, Y.D. Nurseha, A.Y. Aprilio. "Ekstraksi Keyframe dengan Entropy Differences untuk Temu Kembali Konten Video berbasis Speeded-Up Robust Feature". *Jurnal Cybermatika*, 2015, 2.2.
- [6]. F.F. Adi, M. Ichwan, Y. Miftahuddin. "Implementasi Algoritma Speeded Up Robust Features (SURF) Pada Pengenalan Rambu-Rambu Lalu Lintas". *Jurnal Teknik Informatika dan Sistem Informasi*, 2017, 3(3).
- [7]. R. Budde, K. Kautz, K. Kuhlenkamp, H. Zölligoven. "Prototyping: An Approach to Evolutionary System Development". *Springer Science & Business Media*, 6 Des 2012.
- [8]. A. Haselhoff, A. Kummert. "On filtering by means of generalized integral images: a review and applications". *Multidimensional Systems and Signal Processing*, 2012, 23(1-2), p. 291-312.
- [9]. F. Dewanti, R. Sumiharto. "Purwarupa Sistem Penggabungan Foto Udara Pada UAV Menggunakan Algoritma Surf (Speeded-Up Robust Features)". *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 5(2), October 2015. p. 165-176.