

Fitur Pengkategorian Otomatis dari Gambar Berbasis Web dengan Metode SURF dan Haar Cascade Classifiers

Andrew Samuel¹, Kartika Gunadi², Justinus Andjarwirawan³

Program Studi Teknik Informatika

Fakultas Teknologi Industri, UK Petra

Jln. Siwalankerto 121-131 Surabaya 60236

Telp. (031)-2983455

m26413079@john.petra.ac.id¹, kgunadi@petra.ac.id², justin@petra.ac.id³

ABSTRAK

Ecommerce merupakan *Market Place* dan merupakan kemajuan teknologi saat ini dimana sekarang banyak digunakan oleh masyarakat luas untuk melakukan transaksi barang-barang yang dibutuhkan karena mudah dan praktis. Berbelanja dengan menggunakan *Ecommerce* juga tergolong sangat murah dan harga sangat bersaing daripada toko konvensional yang lebih mahal karena terdapat biaya-biaya operasional yang besar. Tetapi kesalahan data yang terdapat pada *Ecommerce* lebih besar daripada sebuah toko konvensional, karena barang tersebut tergantung dari masing-masing orang yang menginputkan.

Dengan semakin berkembangnya teknologi, terutama pada *Web Service* yang membantu *Ecommerce* yang semakin canggih, tentunya terdapat solusi untuk mengatasi permasalahan dan mengurangi kesalahan input yang tentunya dapat merugikan kedua belah pihak. Salah satunya adalah kecerdasan komputer yang dikembangkan untuk dapat mendeteksi dan mengenali suatu benda.

Pendeteksian dan pengenalan obyek yang ditambahkan dalam fitur *Ecommerce* menggunakan metode yang sering digunakan untuk pendeteksi muka yaitu *Haar Cascade Classifier*, serta pengenalan dengan menggunakan *SURF*. Sehingga fitur tersebut dapat meningkatkan performa *Ecommerce* dan tidak menggunakan biaya dan *upgrade* yang berarti.

Kata kunci

Website, Web Service, Ecommerce, Pendeteksian Obyek, Pengenalan Obyek, Haar Cascade Classifiers, SURF, Feature Match.

ABSTRACT

Ecommerce is a Market Place and a technological advancement nowadays where it is now widely used by the public to conducts transaction of needed goods due to very practical. Shopping by using Ecommerce also cheaps and the price is very competitive then conventional store, which more expensive due to large operational costs. But the errors of data contained in Ecommerce is greater than a conventional store, because the data input is depends on the person who using it.

With the development of technology, especially on Web Services that helps Ecommerce grow significantly, of course there are solutions to solve the problem and reduce an errors that would made disadvantageous on both side. And one of them is a Computer Intelligence developed to be able to detect and recognize an object.

The detection and recognition of objects which will be added on Ecommerce for its feature use frequent of image processing methods for face detection, the Haar Cascade Classifier, and the recognition using SURF. So these feature can improve the performance of Ecommerce so as not to use significant costs and upgrade.

Keywords

Website, Web Service, Ecommerce, Object Detection, Object Recognition, Haar Cascade Classifiers, SURF, Feature Match.

1. PENDAHULUAN

Salah satu teknologi yang semakin berkembang yang memberikan dampak positif bagi kehadiran manusia dan peradaban teknologi baru adalah proses visual komputer dalam hal pengolahan gambar (*image processing*). Kata *image* yang berarti gambar, memiliki banyak kegunaan dalam kehidupan sehari-hari. Gambar memberikan suatu informasi, interpretasi, ilustrasi, evaluasi, komunikasi, dan hiburan untuk manusia. Dengan adanya gambar, akan membantu manusia dalam menganalisa, menghafalkan, dan membantu mengingat akan suatu hal. Dengan adanya pengolahan pada gambar, akan membantu memudahkan untuk lebih mengolah pada suatu informasi yang ada.

Pendeteksian Obyek pada gambar (*Object Recognition*) merupakan bagian dari citra visual komputer yang berfungsi untuk mendeteksi sebuah atau banyak obyek pada suatu gambar sehingga dapat mengklasifikasi obyek tersebut menjadi suatu data yang dapat kita terima. Dengan menggabungkan situs website dengan citra visual komputer menjadi suatu aplikasi akan membuat dampak baik pada platform *website* sehingga nilai jual akan semakin tinggi karena masyarakat zaman sekarang banyak yang menggunakan internet untuk kebutuhan mencari informasi tidak hanya teks atau tulisan tetapi juga banyak yang mencari kebutuhan informasi akan gambar. Selain itu, banyak pengguna internet atau aplikasi yang menginginkan situs website atau aplikasi yang dapat memudahkannya dalam mengisi informasi dan bahkan menginginkan situs web atau aplikasi tersebut dapat mengisi sendiri informasi sehingga tidak merepotkan para pengguna.

Untuk melakukan pendeteksian obyek pada gambar, dapat menggunakan banyak metode pada citra visual komputer. Sehingga diperlukan suatu pengujian untuk dapat mendeteksi dan mengenali sebuah obyek dengan metode *Haar Cascade* dan *SURF*.

2. LANDASAN TEORI

2.1 Pengertian Web Server

Web Server adalah sebuah software yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal dengan nama *web browser*[3]. Umumnya *Web Browser* berbentuk dokumen HTML. Sederhananya Web Server mengirimkan data berupa HTML kepada permintaan server sehingga akan terlihat seperti umumnya yaitu sebuah tampilan website. Fungsi utama dari web server yaitu melakukan transfer atau akan transfer berkas permintaan pengguna melalui protokol komunikasi yang telah ditentukan sedemikian rupa. Protokol komunikasi meliputi berkas file, teks, video, gambar, dan masih banyak lagi. Pemanfaatan web server berfungsi untuk mentransfer seluruh aspek file berdasarkan permintaan / *request* dari server untuk mengolah atau *mengeksekusi* file dan mentransfer file dalam suatu pemberkasan file yang berupa file, teks, gambar, dan video.

2.2 Pengertian Visual Komputer

visual komputer adalah bidang yang mencakup metode untuk memperoleh, mengolah, menganalisis, dan memahami gambar. Secara umum, data dimensi yang ada di dunia nyata sangat bermanfaat untuk dapat menghasilkan informasi yang meliputi numerik dan atau simbolik, misalnya dalam bentuk keputusan[4].

Peneliti di seluruh dunia juga sekarang sedang menerapkan untuk membangun lebih robot atau program dengan menggunakan visual komputer dan kecerdasan buatan. Visual komputer terus menerus dikembangkan dengan memasukkan berbagai macam teknik seperti rekayasa, matematika, fisika, dan statistika[1].

2.3 Pengertian Object Detection

dalam tiga tahun terakhir, kemampuan klasifikasi objek dan pendeteksian obyek telah mengalami kemajuan karena meningkatnya teknik visual komputer dan kecerdasan buatan beserta jaringan konvosional[6]. Salah satu berita yang menggembirakan adalah bahwa sebagian besar dari kemajuan bukan dari hardware yang lebih kuat, dataset yang lebih besar, model yang lebih besar, tetapi dari ide-ide yang lebih baru. Sehingga konsep pendeteksian obyek lebih maju karena pemikiran para pengembang yang lebih luas sehingga menerapkan ide yang lebih baik dan efisien dalam pendeteksian obyek.

Pendeteksian obyek dalam hal yang alamiah, memainkan peranan yang sangat penting dalam visual komputer. Metode pendeteksian obyek lebih banyak tentang *local-feature extract* dan *supervised learning*. Dengan tujuan utama dari dulu dari visual komputer adalah membangun suatu sistem yang mampu mengenali berbagai macam benda atau obyek dari gambar yang memiliki gambar yang berantakan (*clustered scene*)[2].

2.4 Pengertian SURF

Algoritma SURF (*Speeded Up Robust Feature*) adalah sebagai algoritma dari visual komputer yang mendeteksi berbagai *interest point* pada kedua gambar yang berbeda dan berulang sehingga mereka dapat secara akurat terdeteksi di berbagai orientasi, skala yang diperbesar atau diperkecil, rotasi dari obyek yang sama[7]. Untuk setiap *interest point*, vektor didefinisikan yang dimana vektor tersebut menyimpan informasi tentang titik-titik didalam daerah obyek tersebut. Setelah itu fitur yang terunik dan terbaik akan dipilih dan

dicocokkan kepada kedua obyek dan kedua gambar. Ini memberikan pemetaan antara fitur dalam gambar-gambar tersebut. Mungkin ada ketidakcocokkan dan ketidaksesuaian titik dengan obyek yang akan dideteksi.

2.5 Pengertian Haar Likes Feature dan Cascade Classifier

Haar-like Feature algoritmanya berasal dari sebuah observasi bahwa setiap daerah daripada gambar memiliki intensitas yang mutlak dan berubah secara drastis nilainya dalam kondisi pencahayaan yang berbeda-beda, tetapi hubungan ordinal dengan objek tersebut tidak berpengaruh nilainya[2]. Algoritma *Haar-Like Feature* terdiri dari dua atau lebih *rectangular* yang digunakan untuk menentukan intensitas pada piksel secara vertikal maupun horizontal yang nantinya akan digunakan untuk menghitung intensitas piksel dari setiap *rectangular* yang ada.

Blok atau *rectangular* ini dapat mengurangi waktu komputasi secara sangat efektif. Algoritma Haar-Likes Feature sama dengan pengurangan dan penjumlahan dari piksel yang berwarna hitam, nilai *grayscale*, dan penjumlahan dari nilai piksel yang berwarna putih[8]. Blok atau *rectangular* yang menghitung ini dapat terdiri dari banyak yang posisinya bisa horizontal dan vertikal

2.6 Pengertian OpenCV

OpenCV merupakan library Image Processing yang dibuat dan dikembangkan oleh Intel. Versi *OpenCV* terbaru adalah versi 2.4.9.1 dan versi 3.1. Library OpenCV adalah *Open Source* dan sangat mudah digunakan dan dapat diinstall dengan mudah. OpenCV juga dirancang untuk efisiensi komputer dalam aplikasi yang *real-time*. Implementasi pertama pada OpenCV adalah dengan menggunakan bahasa pemrograman C. Namun, seiring bertambahnya waktu popularitas yang menggunakan bahasa C berkembang dengan menggunakan bahasa C++. Begitu pula dengan OpenCV yang implementasinya dapat digunakan di bahasa pemrograman C++ di versi 2.0 dengan beberapa fungsi baru. Namun saat ini, *library* OpenCV dapat digunakan dengan bahasa pemrograman selain C dan C++ yang antara lain *Java*, *Python*, dan *Matlab / Octave*. OpenCV tersedia secara bebas dan gratis untuk di download di situs opencv.org.

Semua kelas dan fungsi pada OpenCV berada pada *namespace cv* dan karena itu maka harus memanggil namespace tersebut dengan cara menambahkan *namespace cv* setelah masuk file header atau dengan menambahkan *cv::* pada awalan setiap fungsi, kelas, dan struktur data di OpenCV.

2.7 Pengertian Bahasa Pemrograman Python

Python adalah bahasa pemrograman multiplatform yang *open source* yang terinterpretasi oleh *typing* yang dinamis dan kuat serta memiliki *library* yang banyak juga seperti struktur data, *multithreading*, *files*, dan jaringan[10]. Fakta bahwa bahasa *python* merupakan bahasa yang mudah ditafsir adalah contohnya bebas dari *compilation*. Bahasa pemrograman yang membutuhkan *compilation* antara lain seperti bahasa C dan C++, dengan adanya kerugian dari bahasa yang membutuhkan *compilation* seperti lebih lambat, *cross-compiling on embedded system*. Bahasa pemrograman *Python* juga digunakan di banyak medan seperti contohnya komputasi ilmiah, sistem administrasi, dan *web development*.

Keuntungan utama yang ditawarkan oleh bahasa pemrograman *Python* adalah memiliki alat simulasi *python* gratis. Memang, bahasa yang *Opensource*, dapat diimplementasikan dengan mudah seperti membuat model *client-server model* untuk di tes, dan pemrograman dengan menggunakan *web browser*. Para pelajar yang menginginkan belajar bahasa *python* dapat menguji dan berlatih di rumah tanpa harus menginstal dan membayar software untuk perhitungan dan simulasi secara dinamis pada suatu sistem.

3. DESAIN SISTEM

3.1 Pemilihan Nama Fitur Aplikasi

Fitur aplikasi ini dibuat dengan nama “Ecomtec” yang berarti “*Ecommerce Object Detection*” yang merupakan fitur aplikasi untuk website terutama *ecommerce* yang berfungsi sebagai *marketplace*, dimana *ecommerce* tersebut memiliki kemampuan untuk setiap user dapat mengupload sendiri gambar untuk keperluan masing-masing.

3.2 Analisa Permasalahan

Metode *Haar-Likes Feature* yang dapat mendeteksi obyek tetapi harus dengan data *training* yang harus dilatih. Semakin banyak data *training*, semakin bagus pula hasilnya. Untuk mengenali obyek dibutuhkan detektor obyek tambahan yaitu *SURF (Speeded Up Robust Feature)* yang digunakan untuk mengenali obyek dengan akurat. Pemilihan dalam menggunakan metode *SURF* ataupun *SIFT* yang memiliki perbedaan di tingkat komputasi dan akurasi.

Permasalahan yang timbul dengan adanya penambahan metode pendeteksian obyek *SURF* adalah sampai sekarang masih belum ada *paper* atau *journal* tentang riset penggabungan metode visual komputer pendeteksian obyek *Haar-Likes Feature* dengan *SURF*.

3.3 Desain Sistem

3.3.1 Gambaran Cara Kerja Sistem Fitur Aplikasi

Fitur aplikasi ini dapat digunakan pada website *ecommerce* yang membutuhkannya. Fitur aplikasi ini dapat ditempatkan di halaman *website* yang terdapat input untuk mengupload file gambar sehingga dapat mengambil obyek dari gambar tersebut secara otomatis. Pemilik website *ecommerce* tersebut dapat memberi fungsi sendiri secara kustom, jika fitur aplikasi ini mendeteksi obyek atau tidak dapat mendeteksi obyek.

Cara kerja daripada fitur aplikasi ini sangat mudah yaitu hanya dengan mengupload gambar. Pertama *user* akan melakukan autentikasi login yang biasa dilakukan oleh *ecommerce* setelah itu *user* akan langsung diarahkan ke halaman *dashboard* atau *index* dan terdapat box UploadFile. Upload File dapat dilakukan dengan cara *drag and drop* sehingga memudahkan *user* juga. Setelah *user* melakukan upload file, *user* akan mendapatkan informasi daripada gambar yang telah diupload tersebut. Data yang diupload akan berupa teks yang dimana merupakan nama daripada obyek yang telah berhasil diklasifikasi beserta *cropped* obyek atau obyek yang telah dipotong.

3.3.2 Sistem Server Ecomtec

Pada fitur aplikasi *Ecomtec*, website yang diinstall dengan menggunakan *Virtual Private Server* yang informasinya sebagai berikut:

- CPU : Intel (R) Xeon (R) CPU E-5-2630-L @ 2.00GhZ
- RAM : 1.00GB
- Storage : 30 GB SSD (Solid State Drive)
- Bandwidth: 2 TB Maximum Transfer / Month

- OS : Debian 8.7 x32
- Lokasi : Singapore

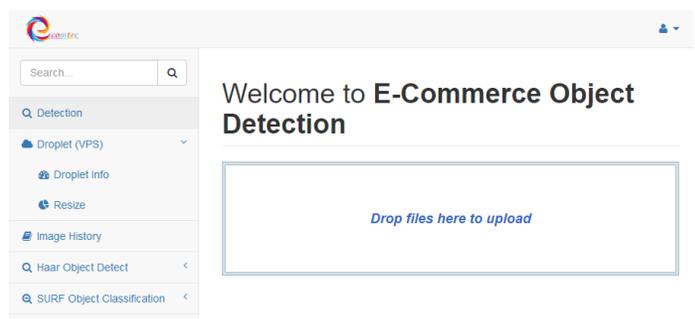
Lokasi yang dipilih adalah Singapore karena provider *Digital Ocean* menyediakan *datacenter* yang terdekat dengan *Indonesia* adalah *Singapore*. Pemilihan *datacenter* yang terdekat merupakan salah satu hal yang sangat penting. Karena pemilihan lokasi menentukan kecepatan akses dalam melakukan suatu pemrograman dalam server tersebut. Jika *Client* dengan *Server* memiliki jarak yang jauh, maka kecepatan akses akan semakin lama karena membutuhkan waktu untuk mencapai ke tempat tujuan, setelah itu server memproses dan mengolah data lalu dikembalikan dalam bentuk *respond* kepada *client* yang juga membutuhkan waktu untuk kembali. Tetapi jika *Client* dan *Server* memiliki jarak yang tidak terlalu signifikan jauh jaraknya, maka akan memotong waktu *request* dan *respond*.

3.4 Desain Interface Ecomtec

3.4.1 Tampilan Dashboard

Setelah berhasil melakukan autentikasi *user* langsung diarahkan ke *pages/index.php* dimana file php merupakan file halaman dashboard. *User* langsung dapat melakukan upload file pada area segi-empat *Drop Files Here to Upload*. Area segi empat merupakan file *input* untuk melakukan upload file yang hanya dapat menerima gambar. File tersebut dikirimkan ke server untuk dapat diproses melalui *Ajax synchronous*, sehingga jika *user* berpindah halaman dan melakukan fungsi lain dapat mengganggu aktivitas proses file *upload* tersebut. Skrip pada *Server* untuk mengolah gambar dan mendeteksi obyek akan dibuat untuk dapat menerima file hanya dengan file gambar dengan format *jpg*, *jpeg*, dan *png*.

Untuk melakukan fungsi pendeteksian obyek pada halaman pertama dengan navigasi bernama *Detection*. Pada navigasi sebelah kiri terdapat beberapa pilihan untuk berpindah halaman yang memiliki fungsi penting masing-masing dalam mendukung pendeteksian obyek antara lain *Droplet VPS*, *Image History*, *Haar Object Configuration*, dan *SURF Database List Image*.



Gambar 1. Tampilan dashboard Ecomtec.

3.4.2 Tampilan Interface Lainnya

3.4.2.1 Informasi Virtual Private Server

Pada halaman ini berisi tentang segala informasi mengenai data *virtual private server / droplet* seperti Nama *Droplet*, kapasitas memory (RAM) yang digunakan, banyaknya CPU yang digunakan, Disk Space yang digunakan, Batasan Transfer Bandwidth per bulan, Tipe OS yang digunakan, Harga setiap jam hingga harga setiap bulannya.

3.4.2.2 List Gambar Database SURF

Pada halaman ini berisi dengan gambar yang terdapat pada *database* dan ditampilkan dengan menggunakan *datatable*. Gambar akan diatur berdasarkan penggolongan kategori gambar. Halaman ini terdapat *datatable jQuery* sebagai modul pengganti tabel untuk memperbaiki desain serta menambah kelengkapan informasi, karena terdapat input *search* untuk mencari gambar. Tabel tersebut mengambil informasi melalui *MySQL* dengan menggunakan *PHP* langsung, lalu diolah menjadi bentuk struktur *datatable*. Informasi yang didapatkan antara lain preview gambar training, nama gambar, dan kategori gambar tersebut yang setiap gambar tersebut digunakan untuk melakukan klasifikasi untuk pengenalan obyek

3.4.2.3 Haar Configuration

Pada halaman ini terdapat fungsi untuk melakukan konfigurasi *Haar Detect Object* dalam sistem *Brute Force* untuk menentukan *Best Match Object*. Karena pada skrip *python* input akan diatur oleh server skrip untuk melakukan pendeteksian secara otomatis agar meningkatkan kinerja *Haar*. Pada fungsi ini terdapat nilai skalar, *Min Neighbor*, *filepath* hasil akhir *Haar* dalam bentuk *jpg* dan *filepath* oleh *Haar XML* sebagai *template* untuk melakukan pendeteksian obyek *Haar Cascade Classifier*.

4. IMPLEMENTASI SISTEM

4.1 Instalasi Server Ecomtec

Pada bagian ini dijelaskan untuk instalasi server pada implementasinya dari desain sistem. Bagian ini menjelaskan instalasi detail agar sistem dapat bekerja dengan lancar.

4.1.1 Inisialisasi Server

Instalasi Sistem Operasi (OS) untuk *Private Server* atau *Droplet* adalah dengan menggunakan Debian v8 x32 (32 bit). Ketika menginstalasi pertama kali sistem operasi, diperlukan langkah-langkah konfigurasi yang harus diambil untuk meningkatkan keamanan pada sistem server dan memberikan dasar yang cukup kuat agar terhindar dari kesalahan atau *error* yang tidak diinginkan.

Semenjak awal instalasi Sistem Operasi Debian v8 x32 bit, biasanya masih tidak ada apapun yang terinstall, sehingga untuk melakukan install, harus dengan melakukan *command sudo*. Untuk dapat melakukan instalasi *software* berikutnya harus terlebih dahulu menginstall komponen untuk memanggil *sudo*. Sebelum awal untuk menginstall *sudo*, pertama kali yang harus dilakukan adalah dengan cara mengupdate server terlebih dahulu agar memiliki komponen yang terbaru dengan menggunakan perintah *apt-get update*. Dengan melakukan satu perintah, semua komponen otomatis akan terupdate yang terbaru sehingga cara tersebut sangat mudah pada *platform linux*. Setelah menunggu sebentar untuk melakukan *download update* pada setiap komponen setelah itu dilanjutkan dengan perintah *apt-get install sudo*.

4.1.2 Instalasi Web Server pada Droplet

Instalasi Web Server merupakan bagian yang penting dimana *private server* atau *droplet* akan memiliki kemampuan untuk menerima, membaca, dan mengolah seluruh informasi yang diminta oleh *client* pada *server* dan direspons oleh *server*. Untuk menginstallasi *Web Server* dengan menggunakan sistem operasi dari *private server* atau *droplet* adalah dengan menggunakan perangkat lunak *LAMP (Linux Apache MySQL PHP, Python, Perl)*.

4.1.3 Instalasi Modul OpenCV

Modul *OpenCV* pada merupakan komponen lainnya pada *virtual private server*. Dapat digunakan untuk *web server* dan dengan bahasa pemrograman lainnya berbasis *linux*. *OpenCV* sangat

berguna untuk melakukan komputasi terhadap gambar dan merupakan *image processing* yang paling populer.

Untuk melakukan instalasi *OpenCV* dapat menggunakan perintah *sudo apt-get build-essential* untuk mempersiapkan kompilasi. Setelah itu menggunakan perintah *sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev*. *Cmake* dan *Git* merupakan komponen yang sangat penting dan krusial untuk *Open CV*. Dengan menggunakan *Cmake*, instalasi *openCV* dapat dikonfigurasi sendiri terhubung modul *openCV* merupakan komponen untuk pemrosesan gambar (*image processing*), kompilasi untuk mengerjakan proses dapat menggunakan *CPU* atau *GPU* dapat dikonfigurasi lewat instalasi secara kustom dari *Cmake*.

4.2 Instalasi Web Ecomtec

Untuk membuat fitur pendeteksian obyek harus dibuat situs *website* terlebih dahulu. Kegunaan dari *website* adalah mempermudah *user* dan pengunjung situs web untuk dapat menggunakannya dengan mudah. Fitur pendeteksian obyek dibuat untuk membantu situs yang mengelola gambar seperti *E-Commerce*. Desain *website Ecomtec* merupakan sebagian daripada desain seperti dasbor pada melihat dasbor pada *website ecommerce* pada umumnya.

4.2.1 File Konfigurasi (Header)

Pada file ini merupakan konfigurasi atau *setting* awal terhadap server yang berisikan tentang bagaimana server tersebut berinteraksi dengan database, *username* dan *password* untuk melakukan hubungan dengan server, setting pada *session*, dan link untuk mengatur *htaccess* agar perpindahan halaman dinamis menjadi lebih mudah.

Pada file konfigurasi header diberi nama *conn.php* yang merupakan konfigurasi awal atau *header* yang akan terdapat pada setiap halaman pada *website Ecomtec*.

4.3 Komponen Fitur Aplikasi Ecomtec

4.3.1 Login Area Ecomtec

Pada halaman ini merupakan tampilan awal untuk *website Ecomtec*. Dengan kombinasi *username* dan *password* yang telah tersedia dalam database untuk dapat masuk ke dalam situs *website Ecomtec*. Hal ini mengadopsi pada situs *website Ecommerce* dimana untuk melakukan upload gambar adalah *user* yang telah mendapatkan identitas daripada situs tersebut lalu menguploadkan gambar untuk keperluan yang terdapat pada situs *website* tersebut.

4.3.2 Halaman Awal (Index)

Halaman awal atau disebut juga dengan *Index* memiliki tampilan seperti *dashboard* seperti *user* biasanya mengakses *user dashboard* dari pada situs *website ecommerce* lainnya, karena untuk melakukan *upload* gambar kebanyakan pengunjung *website* harus memiliki *ID* atau sesi untuk dapat mengetahui informasi tersebut setelah itu dapat mengupload gambar untuk diolah lagi lebih lanjut, seperti gambar tersebut digunakan sebagai *picture* atau *thumbnail* untuk barang jualan. Tema awal pada *Ecomtec* tidak berbeda dengan ada yang di desain sistem.

Sistem *Upload Ecomtec* dengan menggunakan *library Dropzone.js*. *Dropzone* sendiri digunakan hanya untuk membuat input file gambar menjadi jauh lebih bagus dengan esensi kebiru-biruan. Selain itu mempermudah *user* untuk mengupload gambar karena dapat menggunakan *Drag and Drop* dan atau diklik langsung ditunjukkan ke direktori untuk mengupload file dengan mudah. Selain itu *Dropzone.js* dapat diinisialisasi dengan mudah dengan memanggil fungsi $\$(variable).dropzone()$. Dimana didalam fungsi *dropzone* dapat memanggil suatu *event* dan

konfigurasi *dropzone* dari pengaturan file yang dapat diambil hingga membuat *event* yang kustom sesuai dengan kebutuhan.

Jika file telah terupload (fungsi '*sukses*' terlempar) maka akan membuat area *dropzone* menjadi tidak terlihat yang terletak pada *div* dengan identifier '*dropzone_area*'. Setelah itu memanggil *div* dengan identifier '*dropzone_image*' dan '*dropzone_output*' untuk mengeluarkan gambar dan output.

4.3.3 Upload dan Proses Gambar (Server Upload)

Gambar yang diupload pada halaman pertama melalui *Dropzone*, akan dirposes oleh *client* dengan menggunakan *JQuery* dan *javascript* untuk dipertimbangkan kriterianya, setelah itu gambar yang diupload akan dirposes oleh server dengan cara memotong gambar atau *cropping*. Setelah itu gambar tersebut akan dikopikan ke kanvas yang baru dengan kanvas yang sudah di *resize*.

File *server-script* untuk mengolah data gambar yang berhasil terupload diberi nama *upload.php*. Fungsi *resize* dan *cropped* diperlukan untuk menstandarkan gambar karena gambar yang diupload oleh *user* tidak selalu sama. Fungsi ini mengambil beberapa input tinggi dan lebar file yang diinginkan, letak file, dan kualitas file yang secara normal disetting menjadi 90 jika file tersebut berformat '*jpeg*'.

Pertama-tama fungsi ini akan mengecek lebar dan tinggi pada file tersebut. Jika tidak berhasil, maka fungsi akan secara otomatis keluar dan memberikan pesan error pada *javascript*. Setelah itu algoritma akan melakukan pengecekan apakah file sudah berformat gambar. Gambar yang diinputkan akan diambil bentuk, warna, dan dikopikan ke kanvas yang baru dengan latar belakang berwarna putih (jika file *png* yang memiliki latar belakang transparan akan terlihat). Sebelum mengkopikan ke kanvas baru, dengan perumusan algoritma pengecekan rasio gambar, dengan cara membagikan lebar maksimum dengan tinggi maksimum, jika lebar lebih besar daripada tinggi, maka akan diketahui besar rasio file, sehingga file yang telah *dicrop* dan *resize* tidak merusak keaslian gambar. Setelah itu file dengan kanvas yang baru akan disimpan dengan format yang sudah ditentukan.

4.3.4 Pendeteksian Obyek (Haar Cascade Classifier)

Karena keterbatasan bahasa pemrograman *PHP* dalam hal pemrosesan gambar dan tidak dapat terintegrasi dengan *OpenCV*, maka, untuk melakukan pemrosesan gambar digunakan *python* yang terintegrasi dengan *opencv*.

Segmen Program 1. Brute Force Haar.

```
<?php
$filecomplete = $root.'server/python/haar-complete.jpg';
$filepath = $root.$file;
if(!file_exists($filepath)){ /* File gambar tidak dapat ditemukan */}
$max_obj_detect = 1;
$scale = $haar_config['START_SCALE'];
$minNeighbor = $haar_config['START_SCALE'];
for($i = 0; $i < 20; $i++){
    $cmd = escapeshellcmd('python '.$component_filepath . ' .
$filepath . ' . $xml_filepath . ' . $scale . ' . $minNeighbor);
    $cmd = shell_exec($cmd);
    $e_cmd = explode(PHP_EOL , $cmd);
    if(count($e_cmd) < $max_output-2){
        $minNeighbor--;
        $cmd = escapeshellcmd('python '.$component_filepath . ' .
$filepath . ' . $xml_filepath . ' . $scale . ' . $minNeighbor);
        $cmd = shell_exec($cmd);
        break; }
    else if(count($e_cmd) <= $max_output){ break; }
```

Segmen Program 1. Brute Force Haar (lanjutan).

```
else{
    if($minNeighbor > 7 + $scale_added){
        $scale_added++;
        $minNeighbor = 5;
        $scale = $scale + $haar_config['SCALE_INCREMENT'];
    }else{$minNeighbor++;}
    $i++;}
}
```

Pada segmen program 1, merupakan coding algoritma *Brute Force Detect* dari file dengan label *image-detect.php*. Fungsi dari *Brute Force Haar* adalah mengatur pemanggilan fungsi dari bahasa *python* dari perintah *shell* untuk memanggil konsol terminal pada *droplet* secara otomatis tanpa harus melakukan *SSH*. Pertama-tama, modul *image-detect.php* akan memeriksa file-file seperti gambar tersebut apakah ada dan bisa dibuka. Setelah itu juga akan memeriksa modul *python* yang terletak pada direktori *python* dengan nama *haar-cascade.py* dan *cascade.xml* yang digunakan sebagai *template* yang telah dibuat dari *Haar Training* melalui fungsi *opencv*.

Setelah memeriksa modul-modul semua dan file dapat dibuka, akan dideklarasikan variabel seperti *scale* awal dan *minNeighbors* awal yang telah terkonfigurasi dengan database. Pada *database* telah dikonfigurasi bahwa nilai *scale* adalah 1.2 dan *minNeighbor* adalah 3. Dapat dikonfigurasi secara manual oleh *user* lagi. Semakin kecil *MinNeighbors* akan membuat gambar yang terdeteksi lebih toleran, dan menimbulkan lebih banyak *false negative*. Sebaliknya semakin besar *MinNeighbors* akan membuat detektor gambar mengurangi tingkat toleransi sehingga mengurangi *false negative*, tetapi juga berdampak hingga meningkatkan *false positive* (obyek gagal terdeteksi). Fungsi tersebut akan melakukan *brute force* nilai *scale* dan *minNeighbors*. Jika menemukan 1 gambar, maka akan berhenti disitu dan menampilkan hasil. Sedangkan jika gambar lebih dari pada satu, maka pendeteksian obyek akan dilakukan lagi dengan parameter *MinNeighbors* ditingkatkan sebanyak *scale_increment* agar mendapatkan gambar lebih akurat. Jika *MinNeighbors* yang ada lebih daripada nilai *minNeighbor_peak*, maka *stage* ditambah, *scale* ditingkatkan sebesar 0.1 poin dan *min neighbors* akan direset kembali ke angka yang telah terinisialisasi sebelumnya ditambah *stage* yang disimpan dengan variabel bernama *scale_added* (untuk menghindari *false positive* karena *scale* akan mempengaruhi efek pendeteksian obyek).

Segmen Program 2. Haar Cascade Classifier

```
obj_cascade = cv2.CascadeClassifier(cascadePath)
image = cv2.imread(imagePath)
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
obj = obj_cascade.detectMultiScale(image, float(scaleMax),
int(minNeighbor))
i = 1
for (x,y,w,h) in obj:
    cv2.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = image[y:y+h, x:x+w]
    print("Object Found: Detected ({}), cropped file has been
saved to {}".format(i, str(cropped_savefile)))
    cv2.imwrite(cropped_savefile, roi_color)
    i+=1
if i > 1:
    print("Found of Total: {} Objects!".format(len(obj)))
    cv2.imwrite(savfile, image)
else:
    print("Found of Total: {} Object!".format(len(obj)))
```

Pada segmen program 2, merupakan cuplikan koding algoritma dari file bernama *haarcascade.py* yang digunakan untuk mendeteksi obyek yang dipanggil melalui perintah otomatis oleh *image-detect.php*. Pada *haarcascade.py* akan mengambil file yang telah dikirimkan dari *image-detect.php*, komponen file *xml* yang telah ditraining, dan input *scale* serta *minNeighbors* yang dapat dikustom.

Dengan bantuan perintah *import cv2*, maka bahasa *python* dapat terintegrasi dengan modul *OpenCV* yang telah terinstalasi. Fungsi *CascadeClassifier* adalah fungsi untuk mengambil sebuah file performat *xml* untuk dibaca file yang telah ditraining sebelumnya. Setelah itu, dari gambar yang diinputkan, akan dideteksi dengan menggunakan fungsi *DetectMultiScale*.



Gambar 2. Pendeteksian obyek dari *Haar Cascade Classifier*.

4.3.5 Pengenalan Obyek (Image Recognition SURF)

Setelah melakukan pendeteksian obyek dengan menggunakan metode *Haar Cascade Classifier* yang dibantu oleh algoritma *Brute Force Haar* dan *Best Match* untuk meningkatkan akurasi *Haar*, file yang telah dikrop akan disimpan pada direktori yang kemudian file gambar tersebut akan digunakan untuk proses pengenalan obyek atau *Image Recognition* untuk dapat mengenali obyek yang telah terdeteksi.

Segmen Program 3. *Surf-recognize.php*.

```
foreach($categories as $category){
    $objects[$category['category_name']] = 0;
    foreach($images as $image){
        $surf_image_path = $root.$image['image_path'];
        if(file_exists($surf_image_path)){
            $cmd = escapeshellcmd('python ' . $component_filepath . ' ' .
            $filepath . ' ' . $surf_image_path );
            $cmd = shell_exec($cmd);
            $num_surf = intval($cmd); /* Convert String to Int */
            $objects[$category['category_name']] =
            $objects[$category['category_name']] + $num_surf; }
        else{continue; }}
    $objects[$category['category_name']] =
    $objects[$category['category_name']] / count($images);}
$value = max($objects);
if($value > 3){
    $key = array_search($value, $objects);
    /* Output "Sukses dan return konklusi hasil pengenalan"*/}
else{ /* Output "Semi Sukses dan pesan skor terlalu kecil"*/ }
exit();}
```

Pada segmen program 3, merupakan cuplikan koding algoritma dari file *surf-recognize.php* yang merupakan file yang berisi algoritma dengan bahasa pemrograman *PHP* yang dimana akan mengolah konfigurasi *SURF* dan pengambilan gambar dari

database. File gambar akan diambil berdasarkan dari *image_category* yang merupakan kategori gambar yang dapat meliputi *Apple iPhone*, *Apple iPad*, *Samsung Mobile*, *Samsung Tab*, dan lainnya. Setelah mendapatkan kategori tersebut, setiap kategori akan diambil gambar yang digunakan sebagai pembandingan dengan menggunakan *looping* yang dapat ditemukan pada segmen kode bagian *foreach*. *Looping* tersebut akan menjalankan program dengan bahasa pemrograman *python surf-cascade.py* yang akan mendapatkan informasi *interest point*, dan *feature matching*. Dari hasil tersebut akan terakumulasi, dan yang paling besar akan ditentukan sebagai hasil kategori terdekat.

Segmen Program 4. *Surf-cascade.py*.

```
surf = cv2.xfeatures2d.SIFT_create(sigma = 0.6, edgeThreshold =
14)
surf2 = cv2.xfeatures2d.SIFT_create(sigma = 0.6, edgeThreshold
= 12)
image = cv2.imread(imagePath)
imageTest = cv2.imread(imageTrainPath)
kp1, des1 = surf.detectAndCompute(image, None)
kp2, des2 = surf2.detectAndCompute(imageTest, None)
FLANN_INDEX_KDTREE = 0
index_params = dict(algorithm = FLANN_INDEX_KDTREE,
trees = 3)
search_params = dict(checks=100)
flann = cv2.FlannBasedMatcher(index_params, search_params)
matches = flann.knnMatch(des2, des1, k=2)
matchesMask = [[0,0] for i in xrange(len(matches))]
for i,(m,n) in enumerate(matches):
    if m.distance < 0.68 * n.distance:
        matchesMask[i]=[1,0]
        correlation+=1
draw_params = dict(matchColor = (0,255,0),
singlePointColor = (255,0,0),
matchesMask = matchesMask,
flags = 0)
print("{0}".format(correlation))
if (correlation) > 0:
    image3 =
cv2.drawMatchesKnn(imageTest, kp2, image, kp1, matches, None, *
*draw_params)
    cv2.imwrite(savfile, image3)
else:
    print("0")
cv2.waitKey(0)
```

Pada segmen program 4, merupakan file *surf-cascade.py* yang merupakan koding dengan bahasa pemrograman *python* untuk mendapatkan informasi *interest point* dengan algoritma *SURF/SIFT* untuk dilakukan perbandingan file yang didapat dari file krop asli dengan file pembandingan dari *database* pada *surf-recognize.php*.

Pada konfigurasi digunakan *SIFT* karena *SIFT* memiliki komputasi yang lebih akurat dibanding dengan *SURF* karena file yang dibutuhkan adalah file kecil yaitu antara 200 hingga 300 piksel, sehingga dimana informasi gambar sangat menurun, sehingga dibutuhkan metode sebelum daripada *SURF* karena jauh lebih akurat. Namun untuk segi waktu, karena file gambar sangat kecil, *SIFT* tidak jauh berbeda dengan *SURF*. Konfigurasi yang digunakan adalah *sigma* yang lebih kecil karena informasi gambar yang sangat kecil, yaitu 0.4 dengan *default value* adalah 1.6. Sedangkan untuk *edgeTreshold* diset ke 6, dimana untuk *default value* adalah 14. Konfigurasi-konfigurasi penggantian nilai tersebut akan membuat pendeteksian *interest point* menjadi lebih

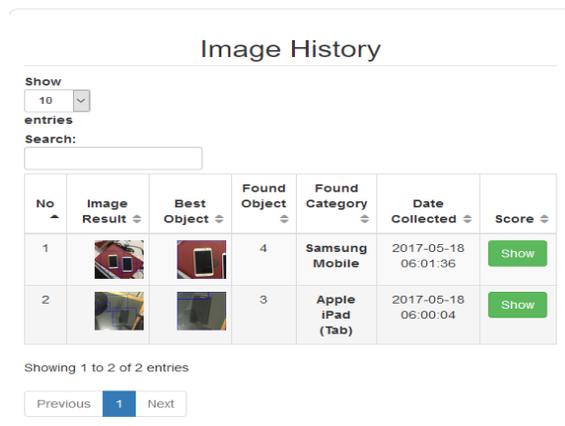
sensitif dan mengurangi *false negative* atau kesalahan *interest point* pada saat pendeteksian obyek dilakukan pada gambar yang resolusinya kurang.

Setelah melakukan konfigurasi, file yang diinput, keduanya akan dideteksi dengan menggunakan fungsi *detectAndCompute* dan hasil variabel daripada *interest point* disimpan dengan nama variabel *kp* dan *des*. Selanjutnya konfigurasi *FLANN* secara *default* dimana *flann_index_kdtree* adalah 0 dan *tree* adalah 3. Setelah melakukan konfigurasi *FLANN*, akan dilakukan *feature matching* dengan menggunakan fungsi *knmatch* dari variabel *des1* dan *des2* yang berisi informasi *interest point* yang didapatkan dari *SURF*. Setelah itu akan mendapatkan *matching*. *Match* tersebut akan ditarik garis dan di filtrasi lagi dengan tingkat kesamaan jarak yang kurang daripada 0.70 akan dianggap benar, selain itu salah. Sehingga hal ini akan mengurangi *false positive* secara signifikan.

4.4 Komponen Lain dari Fitur Aplikasi Ecomtec.

4.4.1 Riwayat Gambar (Image History)

Fitur lain yang terdapat pada fitur aplikasi *Ecomtec* adalah riwayat gambar (*image history*). Dengan menggunakan riwayat gambar diharapkan *user* dapat melihat data kembali dari gambar yang telah dideteksi dengan fitur aplikasi tersebut. Dengan membandingkan dan mencocokkan hasil, dapat meningkatkan kualitas pendeteksian obyek, yang nilai-nilai yang digunakan dalam metode *Haar Cascade Classifier* dan *SURF Cascade* juga tersimpan dalam data riwayat gambar.



Gambar 3. Tampilan riwayat gambar.

4.4.2 Haar Configuration

Haar Configuration merupakan halaman untuk melakukan *setting* beberapa konfigurasi yang digunakan untuk mendeteksi obyek pada gambar. Dengan meningkatkan nilai yang ada pada konfigurasi dapat mempengaruhi hasil pada pendeteksian obyek. Pada halaman ini terdapat didalam navigasi yang bernama *Haar Object Detection*.

4.4.3 SURF Database Image List

Pada komponen ini terletak pada halaman dengan nama *surf-image-list.php* yang menyediakan informasi tentang gambar-gambar yang telah diupload untuk kepentingan proses *SURF Cascade*. Setiap obyek yang telah terdeteksi akan diperbandingkan dengan gambar yang terdapat pada *database* untuk didapatkan skor nya.

5. KESIMPULAN

Pada penelitian berdasarkan dari aplikasi yang telah dikembangkan dan diuji, dapat disimpulkan bahwa baik dari segi *Hardware*, maupun *software* telah memungkinkan untuk dapat mendeteksi obyek dan mengenali dengan metode yang biasa digunakan untuk mendeteksi sebuah muka, dan melakukan *matching* dengan menggunakan metode yang biasa digunakan yaitu *Haar Cascade Classifier* dan *SURF*. Dengan sedikit penambahan dan modifikasi algoritma yang ada serta mengganti *training set* gambar positif dan negatif pada *Haar*, dan *training set* data yang digunakan untuk *SURF* dapat meningkatkan pendeteksian dan pengenalan obyek yang ada. Namun pendeteksian dan pengenalan obyek yang dilakukan harus dalam keadaan tertentu dengan obyek yang tampak terlihat jelas dengan pencahayaan yang cukup.

6. REFERENCES

- [1] A.L. Yuille. 2012. Computer Vision Needs a Core and Foundation. *Image and Vision Computing no.30*, pp. 469-471.
- [2] Alban Vergnaud, Jean Baptise Fasquel, Laurent Atrique. 2015. Python based Internet Tools in Control Education. *International Federation of Automatic Control*, pp.43-48.
- [3] Christian Szegedy et al. 2015. Going Deeper with Convolutions. *University of North Carolina, Chapel Hill, University of Michigan, Ann Arbor, Magic Leap Inc.*
- [4] Danijela Vukadinovic, Maja Pantic. 2015. Fully Automatic Facial Feature Point Detection Using Gabor Feature Based Boosted Classifiers. *IEEE International Conference on System, Man and Cybernetics*.
- [5] Divya Chhabra, Dr. Amandeep Verma. 2012. Multiple Object Detection for Smart TV Shopping Video using Point to Point feature based SURF Method. *Department of Computer Science and Engineering*.
- [6] Er. Saurabh Walia, Er. Satinderjit Kaur Gill. 2014. A Framework for Web Based Student Record Management System using PHP. *International Journal of Computer Science and Information Technology*, vol. 3, issue. 8, pp. 24-33.
- [7] G.M Zhu, Z.L. Ying, dan L.W Huang. 2015. A New Face Recognition Algorithm Based on Haar-Like Features and SRC with Gentle Adaboost. *International Conference on Artificial Intelligence Engineering*.
- [8] Kanuengnit Patoommakesorn, Frédéric Vignat, dan François Villeneuve. 2016. A New Straight Line Matching Technique by Integration of Vision-based Image Processing. *Procedia CIRP Conference no.41*, pp.777-782.
- [9] Kie-Yeong Park, Sun-Young Hwang. 2014. An Improved Haar-Liked Feature for Efficient Object Detection. *Pattern Recognition Letters no.42*, pp. 148-153.
- [10] Krupali Mistry, Avneet Saluja. 2016. An Introduction to OpenCV using Python with Ubuntu. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (2456-3307)*, vol. 1, issue 2, pp. 65-68.
- [11] Silica Kole et al. 2015. SURF and RANSAC: A Conglomerative Approach to Object Recognition. *International Journal of Computer Applications (0975 – 8887)*, vol. 109, no.4

