ARTIKEL

SISTEM PENGHITUNG KENDARAAN DI JALA RAYA MENGGUNAKAN METODE *BLOB TRACKING*

Oleh:

YUNITA SURYA DEWI

14.1.03.02.0214

Dibimbing oleh :

1. **Siti Rochana, M. Pd.**
2. **Julian Sahertian, S. Pd., M.T.**

PROGRAM STUDI

FAKULTAS

UNIVERSITAS NUSANTARA PGRI KEDIRI

TAHUN

SURAT PERNYATAAN
ARTIKEL SKRIPSI TAHUN 2018

Yang bertanda tangan di bawah ini:

Nama Lengkap	:Yunita Surya Dewi
NPM	:14.1.03.02.0214
Telepun/HP	:085852502820
Alamat Surel (Email)	:Tatayunita18@gmail.com
Judul Artikel	:SISTEM PENGHITUNG KENDARAAN MENGGUNAKAN METODE *BLOB TRACKING*
Fakultas – Program Studi	: Fakultas Teknik – Teknik Informatika
Nama Perguruan Tinggi	: Universitas Nusantara PGRI Kediri
Alamat Perguruan Tinggi	: Jl. K.H. Achmad Dahlan No.76 Telepon (0354) 771503 Kota Kediri

Dengan ini menyatakan bahwa :

a. artikel yang saya tulis merupakan karya saya pribadi (bersama tim penulis) dan bebas plagiarisme;

b. artikel telah diteliti dan disetujui untuk diterbitkan oleh Dosen Pembimbing I dan II.

Demikian surat pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari ditemukan ketidaksesuaian data dengan pernyataan ini dan atau ada tuntutan dari pihak lain, saya bersedia bertanggungjawab dan diproses sesuai dengan ketentuan yang berlaku.

Mengetahui		Kediri, 09 Agustus 2018
Pembimbing I	Pembimbing II	Penulis,
Nama : Siti Rochana, M. Pd. NIP / NIDN : 0703018704	Nama : Julian sahertian, S. Pd., M. T. NIP / NIDN : 0707079001	Nama : Yunita Surya D. NPM : 14.1.03.02.0214

SISTEM PENGHITUNG KENDARAAN DI JALAN RAYA MENGGUNAKAN METODE *BLOB TRACKING*

Yunita Surya Dewi
14.1.03.02.0214
Fakultas Teknik – Teknik Informatika
Tatayunita18@gmail.com
Siti Rochana, M. Pd. dan Julian Sahertian, S.Pd., M. T.
UNIVERSITAS NUSANTARA PGRI KEDIRI

ABSTRAK

Banyak terjadinya kemacetan di jalan raya adalah masalah kini yang dihadapi. Salah satunya terjadi di Kota kediri. Mobilitas penduduk yang tinggi, semakin bertambahnya jumlah kendaraan setiap tahun dan minimnya pembangunan infrastruktur jalanan menambah parahnya kemacetan yang ada. Informasi tentang kondisi kepadatan jalan raya tentu sangat dibutuhkan oleh pihak terkait guna memaksimalkan pelayanan penggunaan jalan raya. Oleh sebab itu penulis terdorong untuk membuat suatu sistem untuk menghitung kendaraan agar Dinas Perhubungan dapat memaksimalkan penggunaan guna mengontrol keadaan jalan raya.

Permasalahan penelitian ini adalah (1) Bagaimana mengimplementasikan program deteksi jumlah kendaraan di jalan raya menggunakan bahasa pemrograman *Visual Basic* dan metode *blob tracking* ? (2) Bagaimana performa sistem deteksi jumlah kendaraan ini menggunakan bahasa pemrograman *Visual Basic* dan metode *blob tracking* ?

Penelitian ini menggunakan metode *Blob Tracking* untuk proses penghitungan kendaraan di jalan raya. Sebelum dilakukan pengenalan, *file* citra dilakukan *preprocessing* terlebih dahulu. Proses tersebut meliputi normalisasi ukuran dan pengambilan nilai histogram yang mengacu pada warna aslinya (RGB).

Dari hasil pengujian yang dilakukan pada sistem ini, hasil untuk penghitungan kendaraan diperoleh dengan jumlah data pelatihan sebanyak 23 jenis kendaraan dan data uji coba sebanyak 26 citra memperoleh persentase 91%. Dari hasil pengujian yang dilakukan dapat disimpulkan bahwa metode *Blob Tracking* dapat digunakan untuk menghitung kendaraan di jalan raya.

KATA KUNCI : Penghitung Kendaraan, *Blob Tracking*

I. LATAR BELAKANG

Di era yang semakin berkembang ini, tingginya tingkat populasi penduduk mulai menjadi masalah khususnya dikota-kota. Salah satunya adalah banyak terjadinya kemacetan dijalan raya. Mobilitas penduduk yang tinggi, semakin bertambahnya jumlah kendaraan setiap tahun dan minimnya pembangunan infrastruktur jalanan menambah parahnya kemacetan yang

ada. Informasi tentang kondisi kepadatan jalan raya tentu sangat dibutuhkan oleh pihak terkait guna memaksimalkan pelayanan penggunaan jalan raya.

Untuk mengatasi masalah ini diperlukan teknologi dengan sistem yang mampu melakukan pemantauan arus lalu-lintas di jalan, melakukan manajemen arus lalu lintas, serta memberi solusi alternatif pemilihan rute untuk pengguna jalan. Tingkat kepadatan jalan raya dapat diperoleh dengan melakukan perhitungan terhadap jumlah kendaraan dan kecepatan rata-rata ruang kendaraan dan dengan satuan kendaraan dan dengan satuan kendaraan dalam satu kilometer (kend/km). Selama ini perhitungan tingkat kepadatan jalan raya masih dilakukan secara manual dan hal ini tentunya sangat tidak efektif. Dan dengan adanya tugas akhir ini penulis berusaha melakukan sebuah perancangan dan Analisis program penghitung kendaraan dijalan raya.

Sebelumnya ditulis dalam sebuah penelitian yang berjudul *Deteksi Jumlah Kendaraan Menggunakan Bahasa Pemrograman C++* oleh Fajar Mit Cahyana (2014: 10), penelitian tersebut menentukan kepadatan rata-rata lalu lintas semua objek yang bergerak sehingga pengamat masi harus membedakan kendaraan satu dengan yang lainnya. Penelitian lainnya *Penghitung Jumlah Mobil Menggunakan Pengolahan Citra Digital Dengan Input Video Digital* oleh Mawaddah Aynurrohmah (2011: 03), penelitian ini membahas perkembangan teknologi transportasi meningkat dengan cepat menyebabkan total kendaraan di jalan padat dan sering hambatan pengalaman. Pada pembangunan jalan raya seperti kendaraan bermotor dibuatlah sistem ini. Dari beberapa uraian diatas menjadikan penulis tertarik untuk membuat suatu sistem "Deteksi Jumlah Kendaraan Di jalan Raya Menggunakan Bahasa Pemrograman *Visual Basic* Dengan Metode *Blob Tracking*". Bukan hanya untuk membantu seorang suvei tetapi sistem ini membantu pengurangan kepadatan dijalan raya. *Blob tracking* digunakan karena memiliki banyak subprogram atau *library* yang dapat dikombinasikan sehingga memiliki banyak berbagai fungsi dalam pemrograman yang berkaitan dengan pengolahan citra digital.

II. METODE

A. *Blob Tracking*

Dalam visi komputer, metode deteksi angka bineri ditujukan untuk mendeteksi daerah dalam citra digital yang berbeda dalam properti, seperti kecerahan atau warna, dibandingkan dengan daerah sekitarnya. Secara informal, gumpalan adalah wilayah dari gambar di mana beberapa properti adalah konstan atau hampir konstan; semua titik dalam gumpalan dapat dianggap dalam arti mirip satu sama lain. Metode yang paling umum untuk mendeteksi blob adalah konvolusi.

Menurut Hidayati (2017: 2), penegrtian blob tracking dijelaskan sebagai berikut:

> *Blob detection* atau *blob tracking* adalah algoritme yang digunakan untuk menentukan suatu grup dari piksel saling berhubungan satu sama lain atau tidak. Metode ini sangat berguna untuk mengidentifikasi objek yang terpisah-pisah pada suatu citra atau menghitung jumlah dari suatu objek pada suatu citra. Pada metode *blob detection*, citra harus diproses dengan metode ambang terlebih dahulu, berdasarkan warna yang akan dideteksi. Setelah itu, citra dengan warna diatas nilai ambang dapat dikategorikan sesuai dengan aturan yang telah ditentukan terlebih dahulu. Misalnya untuk piksel yang memiliki nilai lebih kecil daripada nilai ambang akan dikelompokkan sebagai komponen dari objek yang dideteksi, sedangkan yang nilainya diatas ambang dikelompokkan sebagai informasi yang tidak penting. Sedangkan untuk mendapatkan titik berat massa adalah dengan merata-ratakan lokasi tiap piksel dengan warna tertentu. Bagi banyak pengolahan citra, mendeteksi objek *low-level* dalam sebuah gambar merupakan hal yang sangat penting. Objek dalam bentuk dua atau tiga dimensi tersebut biasa disebut *blob*. Bentuk *blob* timbul dalam cara yang berbeda, bergantung pada ukuran, dan dapat dideteksi dengan menggunakan metode sederhana dalam sebuah representasi

B. *Euclidean Distance*

Metode *Euclidean distance* yaitu metode klasifikasi tetangga terdekatnya dengan menghitung jarak antara dua buah objek, perbandingan ini dapat dilakukan dengancara menghitung jarak *Euclidean (Euclidean distance),* yang merupakan selisih antara2 buak vector yang akan dibandingkan untuk

pengenalan sebuah objek yang akandiuji. Metode *Euclidean distance* ini sangat baik untuk pengenalan, metode ini disebut juga jarak *Euclidean*

Menurut Ayu (2017: 40), tahap *euclideaan distance* ini didefinisikan:

> Merupakan tahap pencocokan. Pada tahap ini dilakukan pencocokan antara citra masukkan dengan citra template dengan menghitung nilai kolerasinya. Untuk algoritma euclidean distance, nilai kolerasi dihitung berdasarkan rata-rata jarak antara dua titik. Ada dua jenis ukuran kecocokan, yaitu ukuran kecocokan relatif dan ukuran kecocokan mutlak. Ukuran kecocokan relatif diprgunakan bila dalam suatu kumpulan citra telah diketahui sebelumnya terdapat tepat satu citra yang mengandung objek serupa dengan citra template. Dalam penelitian ini ukuran kecocokan yang dilakukan adalah ukuran kecocokan mutlak sehingga bisa mendapatkan hasil akhir.

Menurut Putra (2010) *euclidean distance* dimaknai sebagai berikut :

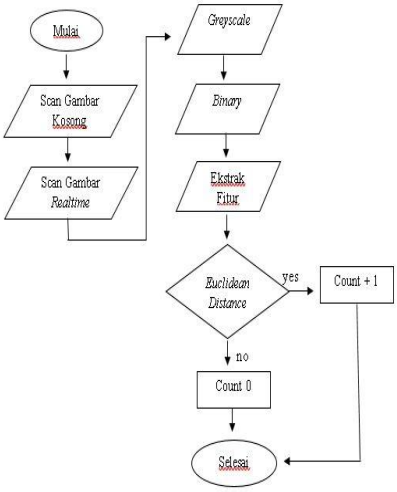
> *Euclidean Distance* adalah matriks yang paling sering digunakan untuk menghitung kesamaan 2 vektor. Jarak *euclidean* menghitung akar dari kuadrat perbedaan 2 vektor (*root of square differences between 2 vector*). Persamaan dari jarak *euclidean* adalah :
>
> $$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \quad \text{..... (2.6)}$$
>
> Hasil perhitungan *euclidean distance* ini akan memperlihatkan seberapa besar tingkat kesamaan antara citra uji dan citra sampel, semakin kecil nilai dari *euclidean distance* (mendekati nilai nol), maka tingkat kemiripan (*similarity*) citra semakin baik.

III. HASIL DAN KESIMPULAN

A. Perancangan *Flowchart* Diagram

Perancangan sistem adalah suatu gambaran alur dari sebuah sistem yang dibangun, dalam skema aplikasi terdapat tahapan – tahapan dalam perancangan sistem agar dapat dipahami sebagaimana alur proses yang berjalan pada sistem yang dibangun. Dalam gambaran perancangan sistem di buatlah suatu bentuk *flowchart* (diagram alur).

Berikut skema pada rancangan sistem penghitung kendaraan dijalan raya dibangun dapat dilihat pada gambar 5.2:



Gambar 5.2 *Flowchart* Penghitung Jumlah Kendaraan

Berikut keterangan yang ada pada gambar 5.2 adalah sebagai berikut :

a. Mulai

Pada tahapan ini sistem menerima input dari kamera deteksi yang sudah terpasang untuk kemudian dapat di proses pada tahap berikutnya.

b. Scan Gambar Kosong

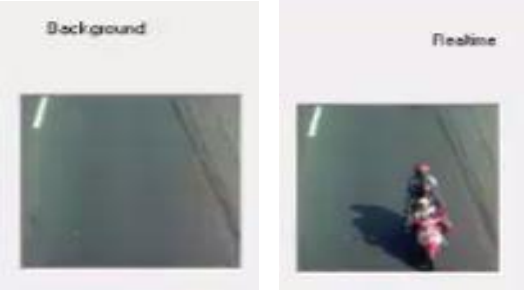
Merupakan langkah awal dari sebuah proses pengolahan citra digital, didalam proses *blob* terdapat tahapan diantaranya menangkap atau mengambil gambar berupa citra digital, merupakan salah satu komponen dari sistem penghitungan jumlah kendaraan yang cukup penting dan mendasar. Tampilan dari proses awal kamera menangkap video ditunjukkan dalam Gambar 5.4

Gambar 5.4 Keadaan awal kendaraan saat belum melintas

c. Scan Gambar Real Time

Tahap dimana kendaraan melewati jalan raya saat itu juga dengan proses konret yang menggunakan media kamera. Input ini dilakukan setiap 5 detik sekali secara terus menerus dan secara otomatis bisa didapat melalui sensor.untuk nilai perbandingan dengan scan gambar kosong akan menghasilkan perbedaan yang berarti ada kendaraan yang lewat. Dalam menggunakan pengolahan *real time* bisa mendapatkan keuntungan banyak karena waktu yang dibutuhkan untuk memperoleh hasil sekejap saja. Jadi tidak perlu menunggu lama untuk mendapatkan hasil yang diinginkan. Dalam pengujian ini dihitung kendaraan yang melintas menggunakan program penghitung kendaraan. Hasil yang didapatkan dibandingkan untuk mendapatkan ketepatan program dalam

menghitung jumlah kendaraan. Pengujian dilakukan dengan menggunakan dua buah gambar. Gambar yang pertama adalah hasil pengambilan gambar dengan keadaan kosong dan gambar kedua yang digunakan adalah hasil lintasan mobil yang melintas di jalan raya. Ditunjukan dalam gambar 5.5.



Gambar 5.5 Perbandingan video saat kosong dan saat kendaraan melintas

d. *Grayscale*

Tahapan proses berikutnya setelah proses *scanning* yaitu proses *Grayscalling*, pengertian dari proses *Grayscalling* sendiri ialah mengubah atau menjadikan citra berwarna RGB menjadi citra berwarna hitam putih abu-abu guna mempermudah tahapan proses berikutnya yaitu perhitungan *pixel.*

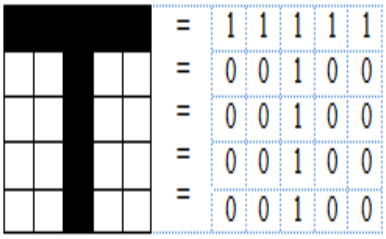
Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra grayscale berbeda dengan citra ”hitam-putih”, dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu ”hitam” dan ”putih” saja. Pada citra grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra grayscale seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spectrum elektromagnetik single band. Berikut adalah contoh proses dari tahapan citra sebelum dan sesudah di lakukan *grayscale* dapat dilihat pada Gambar 5.6:

Gambar 5.6 hasil dari hasil citra RGB sesudah dilakukan proses *greyscale*

e. *Binary*

Citra dimana piksel–pikselnya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (background) dan 1 menyatakan warna tinta/objek (foreground) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra threshold

sebelumnya. Gradasi citra biner dapat dilihat pada Gambar 5.7.



Gambar 5.7 Gradasi Citra Biner

Contoh sebuah citra biner seperti pada Gambar 5.8

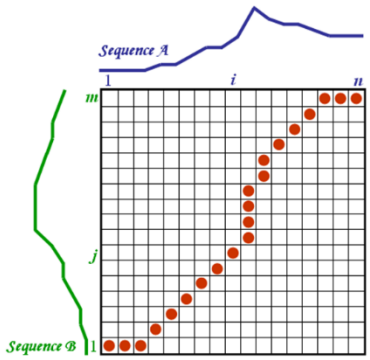
Gambar 5.8 Gradasi Citra Biner

f. *Ekstrak* Fitur

Setelah citra uji yang dikonversi menjadi citra biner, proses selanjutnya adalah proses ekstraksi ciri. Objek-objek yang terdapat pada citra uji tersebut kemudian diekstraksi ciri-cirinya, sehingga tiap-tiap objek tersebut akan muncul ciri-cirinya masing-masing, ini adalah tahap dimana citra biner dibandingkan setelah objek melewati sistem sehingga menimbulkan perbandingan di*background* tersebut. Dengan mendapatkan kesimpulan jika objek tersebut melewati akan muncul nilai. Proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek didalam citra perlu dideteksi seluruh tepinya, lalu menghitung properti-properti objek yang berkaitan sebagai ciri.

g. *Euclidean* Distance

Merupakan tahap pencocokan. Pada tahap ini dilakukan pencocokan antara citra masukkan dengan citra template dengan menghitung nilai kolerasinya. Untuk algoritma euclidean distance, nilai kolerasi dihitung berdasarkan rata-rata jarak antara dua titik. Ada dua jenis ukuran kecocokan, yaitu ukuran kecocokan relatif dan ukuran kecocokan mutlak. Ukuran kecocokan relatif dipergunakan bila dalam suatu kumpulan citra telah diketahui sebelumnya terdapat tepat satu citra yang mengandung objek serupa dengan citra template. Sedangkan ukuran kecocokan mutlak dipergunakan bila dalam suatu kumpulan citra tidak diketahui berapa jumlah citra yang mengandung objek dari salah satu citra template. Dalam penelitian ini ukuran kecocokan yang dilakukan adalah ukuran kecocokan mutlak sehingga bisa mendapatkan hasil akhir. Grafik dari tahap *Euclidean distance* ditunjukan pada gambar 5.9.



Gambar 5.9 Grafik dari tahap *euclidean distance*

h. *Selesai*

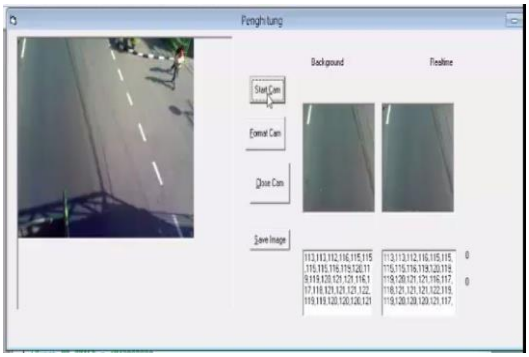
Sistem Selesai

B. Tampilan Program

Pada aplikasi pengenalan motif batik Kediri tampilan program dibuat dengan desain yang sederhana dengan tujuan untuk mempermudah penggunanya. Berikut tampilan program yang dibuat :

1) Tampilan *Splash*

Splash adalah *form* yang pertama kali muncul saat program dijalankan. Tampilan *splash* dapat dilihat pada gambar 5.10:



Gambar 5.10 : Tampilan *Form Splash*

Dari gambar 5.10, tampilan pada halaman *splash* terdapat sebuah *picture box*, fungsi dari elemen ini hanya menampilkan gambar tampilan awala sistem agar lebih menarik.

2) Tampilan *Form* Penghitung

Penghitung adalah *form* yang dapat diakses dari *form splash*. *Form* ini digunakan sebagai akses untuk memproses video. Tampilan pengitung dapat dilihat pada gambar 5.11:

Gambar 5.11 : Tampilan *Form* Penghitung

C. Skenario Uji Coba

Dari hasil uji coba kendaraan yang melintas sesuai dengan kecepatan didapatkan tingkat akurasi yang berdasarkan rumus dibawah ini:

$$\text{Accuracy} = \frac{\text{Jumlah berhasil}}{\text{Jumlah uji coba}} \times 100\% = n\ \%$$

Hasil yang didapat seperti yang terlihat pada tabel 5.1 sebagai berikut..

Tabel 5.1 : Akurasi Skenario Uji Coba

No.	Data	Benar	Salah	Akurasi
1.	Jenis Motor	6	2	6*100/8= 75%
2.	Jenis Mobil	8	0	8*100/8= 100%
3.	Jenis Truk	6	0	6*100/6= 100%
4.	Jenis Sepedah	1	0	1*100/1= 100%
Total		21	2	
Rata – rata Akurasi				21*100/23= 91%

Sesuai data di atas metode ini cocok untukmenghitung kendaraan karena hasil prosentase yang diperoleh 91%.

D. Kesimpulan

Dari hasil penelitian, perancangan, pembuatan dan pengujian sistem penghitung kendaraan di jalan raya menggunakan *Blob tracking* yang telah dilakukan dapat diperoleh kesimpulan sebagai berikut :

1. Untuk mengimplementsi program deteksi kendaraan menggunakan metode *blob tracking* alat utama yang memberikan inputan adalah kamera deteksi yang terpasang pada papan reklame untuk kemudian kamera tersebut mengambil gambar gerak oleh sistem untuk diproses melalui beberapa tahapan scan gambar, grayscalling, untuk diperoleh berupa gambar biner atau berwarna hitam putih yang akan dilanjutkan dengan proses penghitung jumlah kendaraan menggunakan metode *blobtracking.*
2. Performa sistem penghitung kendaraan ini yaitu dengan melakukan uji coba pemasangan kamera di sudut jalan agar tampak semua bagian jalan raya. Setelah itu sistem ini pengambilan objek dipasang tepat pada objek memiliki peran penting dalam proses pengambilan video. Sistem ini lebih akurat karena memperoleh prosentase hasil dari uji coba sebesar 91%.

E. SARAN

Dari hasil penelitian, perancangan, pembuatan dan pengujian sistem penghitung kendaraan di jalan raya menggunakan metode *blob tracking* didapatkan saran untuk penelitian selanjutnya sebagai berikut :

1. Penelitian dilakukan dengan kasus yang berbeda dari

penghitungan kendaraan dijalan raya.

2. Menaruh dibeberapa titik di jalan raya atau titik dimana banyak terjadinya kemacetan.
3. Penelitian selanjutnya dapat menggunakan metode yang lain guna memperoleh hasil yang lebih baik bila dibandingan dengan metode yang digunakan pada penelitian ini.

VI. DAFTAR PUSTAKA

Adi, Dika. 2004. *Identifikasi Objek Berdasarkan Bentuk Dan Ukuran*. Universitas Diponegoro

Aynurrohmah, Mawadah. September 2011. *Penghitung Jumlah Mobil Menggunakan PengolahanCitra Digital Dengan Input Video Digital*. Yogyakarata: STMIK

AMIKOM. Vol. 12 No. 3Ayu, dasriani. 2017. *Perbandingan Pengenalan Karakter Plat Kendaraan Bermotor dengan Metode City Block Distance, Euclidean Distance dan Canberra Distance* TEKNOMATIKA. Vol. 07, No. 02

Balza, Achmad. 2005, *Teknik Pengolahan Citra Digital menggunakan DELPHI*. Yogyakarta : Andi Publishing.

Fausett, L. 1994. *Fundamentals of Neural Network (Architectures, Algorithms, And Applications)*. New – Jersey : Prentice – Hall.

Putra , Darma. 2010. *Pengolahan Citra. Yogyakarta*: CV. ANDI Offset 20.

Haryanti, Sri dan Tri Irianto. 2011. Rancang Bangun Sistem Informasi *E – Commerce* untuk Usaha Fashion. *Jurnal Speed*, 9 (2) : 108 – 114.

Hidayati, Qory. 2017. *Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection*. JNTETI. Vol. 06, No. 02

Kadir, Abdul dan Adhi Susanto. 2013. *Teori dan Aplikasi Pengolahan Citra.* Yogyakarta : Andi.

Kurniadi, Adi. 2000. *Pemrograman Microsoft Visual Basic 6.0.* Jakarta : Elex Media Komputindo.

Kurniawan, Agus. 2013. *Aplikasi Absensi Kuliah Berbasis Identifikasi Wajah*

Menggunakan Metode Gabor Wavelet. Surabaya: Institute Teknologi Sepuluh November

Kusumadewi, Sri, dan Sri Hartati. 2006. *Neuro – Fuzzy Integrasi Sistem Fuzzy dan Jaringan Syaraf*. Yogyakarta : Graha Ilmu.

Samopa, Febrilian. Dan Yulianawati. Juli 2002.*Penerapan Euglidean Distance Pada Pencocokan Pola Untuk Konversi Citra Ke Teks*. ITS

Stefano. 2014. *Cara Membangun Sistem Informasi Mengunakan VB.Net dan Komponen Dxperience*. Yogyakarta: C.V Andi Offset.

Sutojo, T., E. Mulyanto, V. Suhartono, O. Nurhayati dan Wijanarto D. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta : Andi.

Gonzalez, R.C., Woods, R.E., 1992, *Digital Image Processing, Addison-WesleyPublishing Company*, USA.