

PENGUKUR KECEPATAN KENDARAAN DI KAWASAN PEMUKIMAN MENGUNAKAN ALGORITMA IMAGE SUBTRACTING BERBASIS OPENCV

M. Kusuma Alfi Fauzi¹, Beny Firman², dan Muhammad Andang Novianta³
Program Studi S1 Teknik Elektro, Fakultas Teknologi Industri
Institut Sains dan Teknologi AKPRIND
Jln. Kalisahak No. 28, Komp. Balapan, Yogyakarta, Indonesia
alfauzii167@gmail.com¹, benyfirman@akprind.ac.id², m_andang@akprind.ac.id³

ABSTRACT

Driving a vehicle at high speed has greater potential for accidents. Must have a high concentration also must have a high response as well. Vehicle speed measuring device reduces the number of accidents and improve discipline culture in traffic. Measurement of Vehicles Speed In Residential Areas Using Image Subtraction Algorithms Based OpenCV in this study will measure the speed of vehicles in 2 directions, the detection method used is the image subtracting method, while the tracking method used by comparing the current location with predicted location calculated from 1 to 5 previous frames. Testing is done by doing analysis on the detection, tracking, and speed counters. Based on test result, the biggest error value at speed 10 Km/h of 14,34%. And the smallest error value at speed 50 Km/h with the percentage error of 1,62%. This is due to the level of specification of hardware.

Keywords: speed, vehicle, traffic, image processing, opencv

INTISARI

Mengendarai kendaraan dengan kecepatan tinggi memiliki potensi kecelakaan yang lebih besar. Selain harus memiliki konsentrasi yang tinggi juga harus memiliki respon yang tinggi pula. Alat pengukur kecepatan kendaraan diperlukan untuk mengurangi angka kecelakaan serta meningkatkan budaya tertib berlalu-lintas. Pengukur Kecepatan Kendaraan Di Kawasan Pemukiman Menggunakan Algoritma *Image Subtracting* Berbasis OpenCV pada penelitian ini akan mengukur kecepatan kendaraan yang melewati kamera secara 2 arah, metode deteksi yang digunakan adalah dengan metode *image subtracting*, sementara untuk melakukan *tracking* objek dengan cara membandingkan lokasi saat ini dengan lokasi prediksi yang dihitung dari 1 sampai dengan 5 *frame* sebelumnya. Pengujian dilakukan dengan melakukan analisis pada bagian deteksi, *tracking*, dan penghitungan kecepatan. Dari hasil pengujian kecepatan nilai *error* terbesar pada kecepatan 10 Km/Jam sebesar 14,34% sementara paling kecil berada pada kecepatan 50 Km/Jam dengan prosentase *error* sebesar 1,62%. Hal ini dipengaruhi oleh tingkat spesifikasi *hardware*.

Kata Kunci: kecepatan, kendaraan, lalu-lintas, pengolahan citra, opencv.

I. PENDAHULUAN

Mengendarai kendaraan memiliki resiko terjadinya kecelakaan. Salah satu penyebab terjadinya kecelakaan pada transportasi darat adalah kecepatan. Kecepatan kendaraan bermotor mempunyai peranan yang penting dalam persoalan luka dari kecelakaan lalu lintas di jalan raya, kecepatan juga mempengaruhi resiko tabrakan dan akibat tabrakan (Sahabudin, et al., 2011).

Pembatasan kecepatan kendaraan telah diatur pada Undang-Undang No. 22 tahun 2009

tentang Lalu Lintas dan Angkutan Jalan pasal 21 no.1–5. Dimana diperkuat oleh Peraturan Menteri Perhubungan Republik Indonesia PM 111 tahun 2015 tentang Tata Cara Penetapan Batas Kecepatan. Penetapan batas kecepatan ini memperhitungkan segala faktor, namun secara umum dapat dibagi menjadi 4 yaitu di jalan bebas Hambatan paling rendah 60 Km/Jam dan paling tinggi 100 Km/Jam, jalan antarkota paling tinggi 80 Km/Jam, Kawasan perkotaan paling tinggi 50 Km/Jam, dan Kawasan Pemukiman paling tinggi 30 Km/Jam (Cahya, 2015).

Perkembangan teknologi saat ini sudah semakin pesat dalam berbagai bidang termasuk sistem transportasi. Dimana dengan adanya teknologi CCTV dan pengolahan citra dapat dengan mudah melakukan identifikasi kendaraan. Ini mempermudah dalam tugas tugas pengawasan dan penindakan terhadap pelanggaran lalu lintas. Dimana pengawasan secara langsung terkendala masalah masalah yang ada di lapangan.

Penelitian yang berhubungan dengan pengukuran kecepatan kendaraan antara lain:

Pada Alat Pendeteksi Kecepatan Kendaraan Bermotor Menggunakan *Mikrokontroler* dan Webcam Berbasis *Personal Computer* (Setianingrum, 2011). Sensor yang digunakan untuk mengukur kecepatan menggunakan dua buah LDR diukur berdasarkan jarak dari awal waktu sampai akhir waktu tempuh yang di dapat. Jarak kedua tempat (s) per waktu yang ditempuh (t). Setelah kecepatan diukur jika kecepatan melebihi 40Km/Jam maka pelanggar akan *dicapture* menggunakan kamera webcam.

Pada penelitian Sistem Pengukur Kecepatan Kendaraan Berbasis Pengolahan Video (Sadewo, 2015), Metode yang digunakan adalah *Background Subtraction* dengan Algoritma Gaussian Mixture Model (GMM) untuk memisahkan *foreground* dengan *background*. Posisi kamera untuk mengambil gambar video diletakan 90° dari kendaraan yang di deteksi. Sehingga hanya dapat mendeteksi satu kendaraan pada saat melakukan pengukuran.

Untuk mengurangi resiko kecelakaan dan akibat kecelakaan karena kecepatan serta kemajuan teknologi pengolahan citra dalam transportasi yang dapat mengurangi pengawasan secara langsung. Maka dapat diperoleh rumusan masalah sebagai berikut bagaimana merancang sistem deteksi dan penghitungan kecepatan kendaraan dengan algoritma *image subtracting* pada *library* OpenCV serta pengaruh waktu pengolahan citra terhadap hasil data kecepatan. Tujuan penelitian ini adalah merancang sistem deteksi pengukur kecepatan kendaraan menggunakan pustaka OpenCV, merancang alat berdasarkan Undang-undang yang berlaku untuk meningkatkan budaya tertib berlalu lintas.

II. METODOLOGI

A. Alat dan Bahan

Adapun alat beserta spesifikasinya yang dibutuhkan pada penelitian ini antara lain:.

Tabel 2.1 Alat Penelitian

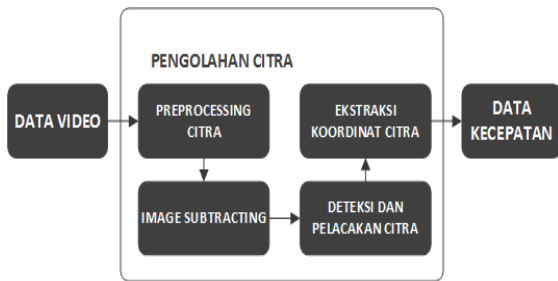
No.	Nama Alat	Keterangan
1.	Notebook ASUS A456UR Spesifikasi : Intel Core i5-7200U, 4 GB RAM, NVIDIA GF 930MX, 1 TB HDD	Menulis, Menjalankan dan Menguji coba sistem
2.	Nikon D3000 kit 18-55mm Spesifikasi : 24 MP CMOS, 1080p / 60 fps Video Recording, 18- 55mm f/3.5-5.6G VR Lens	Mengambil video lalu lintas
3.	Tripod	Dudukan kamera
4.	Visual Studio 2015	Menulis Program
5.	Toyota Calya 2017 Dimensi (P x L x T) : 4070 x 1600 x 1655 (mm)	Sebagai objek yang diukur kecepatan

Adapun bahan yang digunakan dalam penelitian ini menggunakan data *video* yang diambil menggunakan kamera Nikon D3000. *Video* lalu lintas diambil di jalan KRT Pringgodiningrat, Beran, Tridadi, Kec. Sleman, Kab. Sleman, Daerah Istimewa Yogyakarta. *Video* diambil dari atas jembatan menghadap kejalan sehingga dapat menangkap kondisi lalu lintas. Adapun *video* yang digunakan terdiri dari 5 *video* kalibrasi dan 1 *video* untuk pengujian sistem keseluruhan pengukur kecepatan.

B. Perancangan Sistem

1. Deskripsi Sistem

Sistem yang dirancang pada penelitian ini berupa perangkat lunak yang dapat mengolah data *video*. Data *video* yang digunakan adalah *video* yang menyorot kondisi jalan yang sedang dilalui oleh kendaraan. Adapun sistem kerja dari alat ini dapat ditunjukkan oleh Gambar 2.1



Gambar 2.1 Data Flow Diagram Pengukur Kecepatan Kendaraan

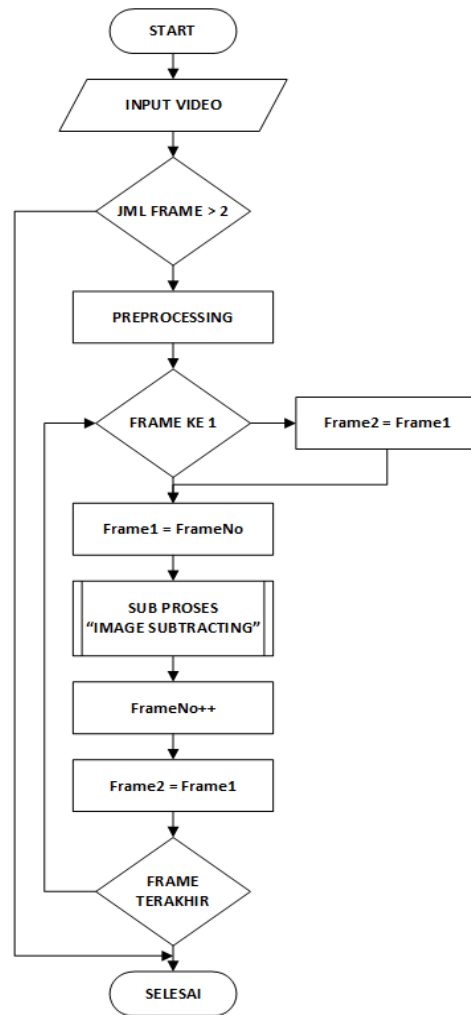
Dari blok diagram diatas, system ini dimulai dari memasukan data video yang akan diolah. Selanjutnya oleh system akan dilakukan *preprocessing* citra. Dalam *preprocessing* ini data citra tiap *frame* akan ditampilkan berurutan agar terlihat seperti gambar yang bergerak. System deteksi kendaraan yang digunakan dengan metode *image subtracting*. Metode ini membutuhkan minimal 2 citra untuk memisahkan antara *background* dan *foreground* atau objek. Adapun cara kerja metode ini yaitu dengan cara mencari perbedaan di antara 2 *frame* citra tersebut.

Ketika objek atau kendaraan yang berhasil terdeteksi dan dapat ditracking melewati titik pengukuran, maka akan di ekstraksi titik koordinat awal dan koordinat akhirnya serta waktu yang ditempuh antara ke dua koordinat tersebut, dari data koordinat awal dan koordinat akhir ini dihitung jaraknya. Untuk waktu yang ditempuh diperoleh dari selisih waktu ketika melewati garis akhir dan ketika melewati garis awal.

Kemudian kecepatan didapatkan dari hasil pembagian antara jarak dibagi dengan waktu yang ditempuh. Sebelum itu dilakukan kalibrasi terhadap jarak dalam video dengan jarak sebenarnya. Sehingga didapatkan kecepatan dengan satuan km/jam.

2. Perancangan Algoritma

Perancangan Algoritma dilakukan untuk memudahkan dalam penulisan progam. Algoritma berisi tentang bagaimana program tersebut akan dijalankan oleh system seperti pada Gambar 2.2.



Gambar 2.2 Flowchart Algoritma Pemrograman

III. HASIL DAN PEMBAHASAN

Hasil pada penelitian ini adalah sebuah aplikasi *GUI (Graphical User Interface)* yang dapat mendeteksi kendaraan dengan menggunakan algoritma *image subtracting* dan menghitung kecepatan setelah melalui titik deteksi. Program tersebut akan menampilkan beberapa jendela dari pengolahan citra tersebut serta menampilkan hasil pengukuran kecepatan dan menyimpan *capture* kendaraan yang melanggar pada *folder* tertentu.

Untuk dapat mengetahui apakah system dapat bekerja sesuai dengan perancangan, maka dilakukan beberapa pengujian. Adapun pengujian tersebut antara lain:

A. Pengujian Deteksi Kendaraan

Pada pengujian deteksi kendaraan ini akan dilakukan pengujian pada tahap-tahap perubahan citra. Dari hasil perubahan tersebut akan di analisis hasilnya sehingga dapat dilakukan perbaikan agar memperoleh hasil yang lebih akurat. Pengujian ini menggunakan *file video* dengan resolusi 1280 x 720 dan *frame rate* 30 FPS.



Gambar 3.1 Frame 1 Citra Asli



Gambar 3.2 Frame 2 Citra Asli

Algoritma *image subtracting* membutuhkan minimal 2 *frame* untuk dapat dijalankan. Sehingga pada *preprocessing* disiapkan 2 buah *frame* citra yang terdiri dari *frame* saat ini (Gambar 3.1) dan *frame* sebelumnya (Gambar 3.2).



Gambar 3.3 Frame 1 Grayscale

Pada Gambar 3.3 dan Gambar 3.4 citra asli dilakukan *grayscale* yang berfungsi untuk

menyederhanakan citra. Sebab, pada citra berwarna terdiri dari 3 macam warna sedangkan citra *grayscale* hanya memiliki 1 warna. Sehingga proses komputasi akan lebih cepat 3:1.



Gambar 3.4 Frame 2 Grayscale

Gambar 3.5 dan Gambar 3.6 berfungsi untuk memberikan efek *Gaussian blur*. Pada tahap ini akan memperhalus detail detail pada suatu citra. Ini bertujuan untuk memperhalus piksel-piksel citra sehingga dapat mengurangi timbulnya *noise* pada tahap tahap selanjutnya.



Gambar 3.5 Frame 1 Gaussian Blur



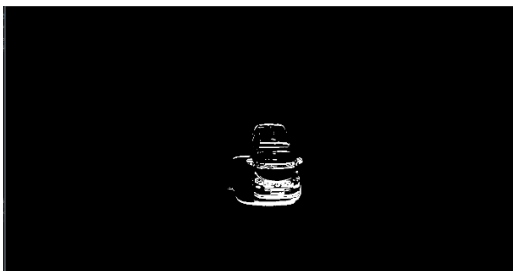
Gambar 3.6 Frame 2 Gaussian Blur

Setelah citra pada *frame 1* dan *frame 2* dilakukan *Gaussian blur* maka tahap selanjutnya adalah tahap *image subtracting*, proses ini akan mencari perbedaan diantara 2 *frame* tersebut. proses *image subtracting* ini akan memperoleh citra baru yang intensitas

setiap pikselnya adalah selisih antara intensitas piksel pada *frame 1* dan *frame 2* (Munir, 2004). Sehingga menghasilkan citra seperti pada Gambar 3.7.

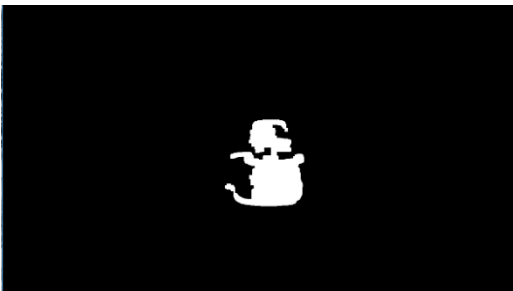


Gambar 3.7 Citra *Image Subtracting*



Gambar 3.8 Citra *Thresholding*

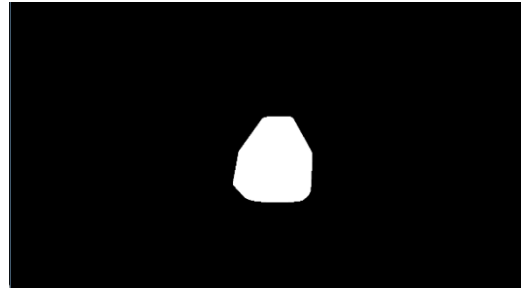
Untuk tahap selanjutnya adalah tahap *thresholding*, Tujuannya agar citra lebih dapat diinterpretasikan oleh komputer. Pada operasi ini piksel-piksel akan dikelompokkan dalam citra berdasarkan batas nilai intensitas tertentu. Pada operasi ini hasil proses suatu titik atau piksel tidak tergantung pada kondisi piksel-piksel tetangganya, hanya tergantung pada kondisi piksel itu sendiri. Proses *thresholding* dapat dilihat pada Gambar 3.8.



Gambar 3.9 Proses Erosi dan Dilasi

Pada tahap *thresholding* citra masih terdapat area yang masih kosong/berlubang. Ini akan menimbulkan masalah pada saat proses deteksi. Untuk itu perlu dilakukan perbaikan terhadap objek melalui erosi dan dilasi. Dimana

erosi digunakan untuk mengurangi piksel-piksel di sekitar objek, sementara dilasi digunakan untuk menambahkan piksel-piksel disekitar objek. Proses erosi juga dapat menghilangkan beberapa *noise* yang kecil.



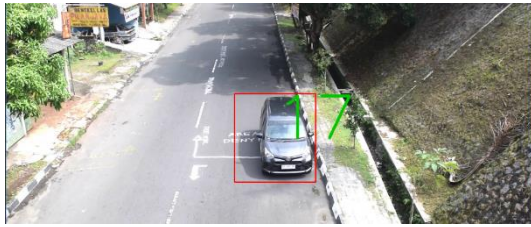
Gambar 3.10 Proses *FindCountour*

Setelah itu dilakukan deteksi objek melalui operasi *findCountour*. Dimana proses ini akan mencari perbedaan piksel antara objek dan background. Agar tidak semua perbedaan piksel dideteksi sebagai objek maka perlu dilakukan *filter*. *Filter* yang dilakukan berdasarkan ukuran objek yang akan dideteksi. Adapun yang menjadi pertimbangan pada *filter* objek dapat dilihat pada Gambar 3.11.

```
for (auto &convexHull : convexHulls) {
    Blob deteksiBlob(convexHull);
    if (deteksiBlob.buatKotak.area() > 400 &&
        deteksiBlob.aspRasio > 0.2 &&
        deteksiBlob.aspRasio < 4.0 &&
        deteksiBlob.buatKotak.width > 40 &&
        deteksiBlob.buatKotak.height > 40 &&
        deteksiBlob.pjgDiagonal > 60.0 &&
        (cv::contourArea(deteksiBlob.bentuk)/
        double)deteksiBlob.buatKotak.area() > 0.50) {
        currentFrameBlobs.push_back(deteksiBlob);
    }
}
```

Gambar 3.11 *Filter* Deteksi Objek

Tahap selanjutnya adalah dengan memberikan kotak (*BoundingRect*) pada objek yang dideteksi. ini bertujuan untuk menandai bahwa terdapat objek yang terdeteksi. Sekaligus kotak ini digunakan untuk mengetahui posisi saat menghitung kecepatan. Aplikasi *bounding rect* dapat dilihat pada Gambar 3.12.



Gambar 3.12 Membuat *BoundingRect*

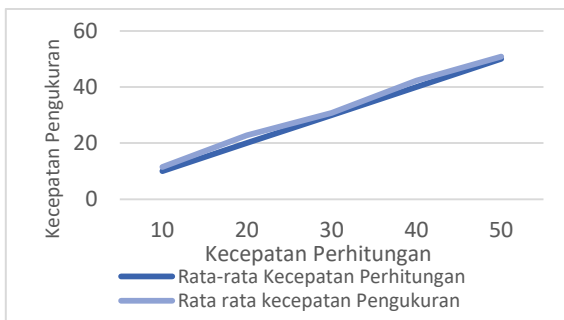
B. Pengujian Data Kecepatan

Pengujian data kecepatan dilakukan untuk mengetahui tingkat linieritas pada setiap tingkat masing-masing kecepatan. Pengujian dilakukan dengan menggunakan *file video* kendaraan yang melaju dari kecepatan 10 Km/Jam hingga 50 Km/Jam. *File video* yang digunakan mempunyai resolusi 1280x720 dan *frame rate* 30 FPS.

Tabel 3.1 Hasil Data Kecepatan Rata-rata

No.	Rata-rata kecepatan Perhitungan (Km/Jam)	Rata-rata Kecepatan Pengukuran (Km/Jam)	Nilai <i>error</i> (%)
1.	10	11,434	14,344
2.	20	22,770	13,85
3.	30	30,786	2,16
4.	40	42,311	5,77
5.	50	50,814	1,62

Kemudian dari masing masing kecepatan tersebut dihitung rata-rata sehingga dapat diketahui rentang *error* yang terjadi. Sehingga dapat digunakan sebagai faktor koreksi.



Gambar 3.13 Grafik Rata-rata Pengujian Kecepatan

Dari Tabel 3.1 Hasil Data Kecepatan Rata-rata yang memiliki tingkat *error* terbesar adalah pada kecepatan 10 Km/Jam. Dimana pada kecepatan 10 Km/jam pendeteksian citra tidak stabil karena jarak perpindahan antar *frame* sangat kecil, sehingga menyebabkan pengukuran kecepatan menjadi memiliki *error*

yang besar pula. Sementara tingkat *error* paling kecil berada pada kecepatan 50 Km/Jam. Ini disebabkan panjangnya citra yang di *subtracting* menyebabkan perbedaan yang tinggi pula, sehingga deteksi dapat lebih akurat. Kemudian tingkat *error* terendah ke 2 berada pada kecepatan 30 Km/Jam. Pada kecepatan ini deteksi juga dilakukan dengan akurat.

Kemudian dari Gambar 3.13 dapat diketahui perubahan kecepatan dari 10 sampai dengan kecepatan 50 Km/Jam dapat dikatakan linier dengan penyimpangan maksimal pada 14%.

C. Pengujian Waktu Pengolahan Citra

Pengujian ini bertujuan untuk mengetahui berapa lama waktu yang dibutuhkan dalam pengolahan citra setiap *frame* dalam tiap detik. Karena waktu dalam pengukuran kecepatan merupakan hal yang penting. Pengujian ini dilakukan dengan menggunakan file video uji performa berdurasi 2 menit 36 detik, resolusi 1280 x 720 dan *frame rate* 30 FPS. Dengan *frame rate* sebesar 30 FPS maka waktu pengolahan citra yang ideal tiap *frame* ditentukan melalui persamaan 3-1.

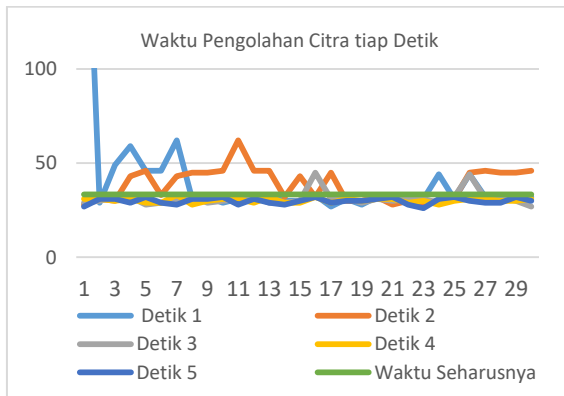
$$\frac{1 \text{ second}}{\text{frame rate}} = \text{waktu pengolahan} \quad 3-1$$

Tabel 3.2 Waktu Pengolahan Citra

Frame ke	Waktu Setiap Detik					Waktu
	1	2	3	4	5	
1	235	28	29	31	27	33,33
2	29	31	30	31	31	33,33
3	49	30	32	30	31	33,33
4	59	43	32	31	29	33,33
5	46	46	28	29	32	33,33
6	46	33	29	29	29	33,33
7	62	43	30	33	28	33,33
8	31	45	32	28	31	33,33
9	32	45	29	30	31	33,33
10	29	46	30	31	32	33,33
11	31	62	31	31	28	33,33
12	30	46	32	29	31	33,33
13	32	46	29	32	29	33,33
14	30	32	29	29	28	33,33
15	30	43	30	29	30	33,33
16	33	32	45	32	32	33,33
17	27	45	31	29	29	33,33
18	31	30	31	31	30	33,33
19	28	30	29	30	30	33,33
20	32	32	31	31	31	33,33
21	31	28	30	31	32	33,33
22	31	30	31	29	28	33,33
23	31	29	31	30	26	33,33
24	44	30	29	28	31	33,33
25	31	31	32	30	32	33,33
26	44	45	44	31	30	33,33
27	32	46	31	31	29	33,33
28	30	45	32	30	29	33,33
29	31	45	30	30	32	33,33
30	32	46	27	31	30	33,33

Rata-rata	41,96	38,76	31,2	30,23	29,93	33,33
-----------	-------	-------	------	-------	-------	-------

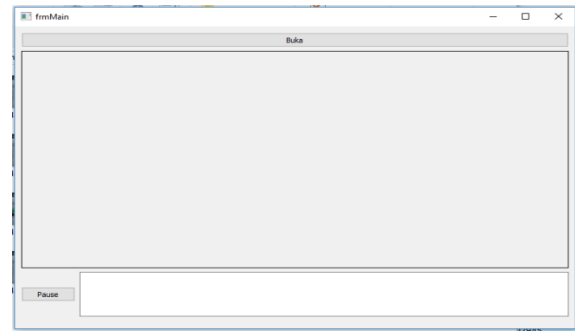
Data dari Tabel 3.2 dapat digambarkan melalui grafik yang ditunjukkan oleh Gambar 3.14 dapat disimpulkan saat video mulai diolah akan membutuhkan waktu lebih lama. Ini karena *frame-frame* yang akan diolah sedang dipersiapkan. Sementara mulai dari detik ketiga dan seterusnya rata-rata waktu pengolahan citra cenderung stabil antara 29 ms hingga 31 ms. Meskipun pada beberapa *frame* terdapat waktu pengolahan citra yang melebihi 40 ms.



Gambar 3.14 Grafik Pengujian Waktu Pengolahan Citra

D. Pengujian Sistem Keseluruhan Pengukur Kecepatan

Tujuan dilakukan pengujian ini adalah untuk mengetahui apakah Pengukur Kecepatan Kendaraan Di Kawasan Pemukiman Menggunakan Algoritma *Image Subtracting* Berbasis OpenCV ini dapat diaplikasikan atau tidak. Pengujian dilakukan dengan menggunakan jendela GUI seperti pada Gambar 3.15 yang dibuat dengan aplikasi Qt Creator 4.2.1 Community. Sementara data pengujian menggunakan *file video* uji performa berdurasi 2 menit 36 detik, resolusi 1280 x 720 piksel dan *frame rate* 30 FPS.



Gambar 3.15 Tampilan Hasil Perancangan GUI Pengukur Kecepatan

File video dimasukkan dengan meng-klik tombol buka kemudian pilih file *video* yang akan diukur kecepatan kendaraan. *File Video* yang digunakan adalah *video* kondisi jalan yang dilewati oleh kendaraan. Kendaraan pada penelitian ini berupa motor dan mobil. Hasil pengukuran tersebut dapat ditunjukkan pada Gambar 3.16. Hasil pengukuran kecepatan kendaraan akan digunakan sebagai data apakah kendaraan itu melanggar atau tidak. Set Point kendaraan yang melanggar adalah kendaraan dengan kecepatan melebihi 30 Km/Jam. Ini berdasarkan PM 111 tahun 2015 tentang Tata Cara Penetapan Batas Kecepatan. Jika kendaraan yang diukur kecepatannya melanggar. Maka sistem akan *capture* gambar kendaraan tersebut pada *folder* dataPelanggar.



Gambar 3.16 Tampilan GUI ketika dijalankan

Tingkat keberhasilan deteksi ini dipengaruhi oleh beberapa faktor. Misalkan pada bus, ukuran bus yang terlalu besar menyebabkan terjadinya multi deteksi. Kemudian pada kendaraan minibus terjadi multideteksi yang dikarenakan dimana posisi mini bus terlalu jauh dari sudut pengambilan gambar.

IV. KESIMPULAN

Berdasarkan hasil dan analisis data pengujian, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Sistem deteksi kendaraan menggunakan metode algoritme *image subtracting* dapat bekerja dengan baik. Algoritme *image subtracting* dapat digunakan untuk mencari objek di dalam sebuah *moving image (video)*.
2. Berdasarkan data pengujian dengan *video* yang mempunyai resolusi (1280 x 720) piksel dan *frame rate* 30 FPS. Pada kecepatan 10 Km/Jam, terdapat banyak kesalahan deteksi. Ini disebabkan piksel yang berpindah sangat kecil seolah-olah piksel tidak bergerak.
3. Sistem pengukur kecepatan dapat bekerja dengan baik dengan tingkat akurasi kesalahan paling kecil 1,62% pada kecepatan 50 Km/Jam, dan yang paling besar 14,34% pada kecepatan 10 Km/Jam.
4. Dari pengujian keseluruhan dengan *video* 2 menit 36 detik didapatkan 28 kendaraan yang berhasil dideteksi, banyaknya deteksi yang berhasil sebanyak 23 dan 5 *error*.
5. Dari pengujian keseluruhan juga didapatkan bahwa pengukuran kecepatan 2 arah dapat dilakukan.
6. Waktu pengolahan citra pada OpenCV tergantung dari tingkat kinerja *hardware (Laptop)*. Semakin tinggi spesifikasi *hardware* semakin cepat pula pengolahan citra.

Saran yang diberikan untuk penelitian selanjutnya yang berhubungan dengan penelitian ini adalah :

1. Pada penelitian selanjutnya dapat memakai *file video frame rate* yang lebih rendah sehingga dapat mengukur kecepatan rendah.
2. Untuk waktu pengolahan yang sesuai dengan durasi *video* yang sebenarnya dapat dilakukan dengan menambahkan waktu tunda. Dengan catatan waktu pengolahan *video* lebih cepat dari pada durasi *video* yang sebenarnya.
3. Dapat ditambahkan sistem identifikasi kendaraan menggunakan *OCR (Optical Character Recognition)* pada plat nomor kendaraan.

V. UCAPAN TERIMAKASIH

Peneliti menyadari bahwa penelitian ini bukan semata-mata hasil kerja keras peneliti pribadi. Oleh sebab itu peneliti menyampaikan ucapan terimakasih kepada semua pihak yang telah membantu memberi dukungan selama penelitian sehingga penelitian ini dapat selesai tepat waktu.

VI. DAFTAR PUSTAKA

- Cahya, K. D. (2015, Agustus 12). *Kompas.com*. Retrieved Juni 13, 2017, from <http://megapolitan.kompas.com/read/2015/08/12/12451781/Polisi.Batas.Kecepatan.Kendaraan.Diatur.UU.Lalu.Lintas>
- Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Sadewo, S. S. (2015). Sistem Pengukur Kecepatan Kendaraan Berbasis Pengolahan Video. *IJEIS*, 5, 177.
- Sahabudin, Wartatmo, H., & Kusčitawati, S. (2011, Juni). Berita Kedokteran Masyarakat. *Pengendara sebagai Faktor Resiko Terjadinya Kecelakaan Lalu Lintas Sepeda Motor Tahun 2010*, p. 94.
- Setianingrum, I. (2011). Alat Pendeteksi Kecepatan Kendaraan Bermotor Menggunakan Mikrokontroler dan Webcam Berbasis Personal Computer. *Jurnal Informatika*, 11, 28.