

## **LINE CROSSING DETECTOR SYSTEM PADA REAL-TIME SITUATIONAL AWARENESS MENGGUNAKAN SPATIAL SAMPLE DIFFERENCE CONSENSUS**

**Salma Meldiyana<sup>1</sup>, Imaniar Salsabila Fathina<sup>2</sup>, Rifqi Yuner<sup>3</sup>, Setiadi Rachmat<sup>4</sup>, Maisevli Harika<sup>5</sup>**

<sup>1,2,3,4,5</sup>Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Bandung  
Email: <sup>1</sup>salma.meldiyana.tif18@polban.ac.id, <sup>2</sup>imaniar.salsabila.tif18@polban.ac.id,  
<sup>3</sup>rifqi.yuner.tif18@polban.ac.id, <sup>4</sup>setiadi@jtk.polban.ac.id, <sup>5</sup>maisevli@polban.ac.id

### **Abstrak**

Pengawasan melalui *Closed Circuit Television* (CCTV) merupakan salah satu antisipasi ancaman pada *Very Important Person* (VIP) dan atau *Very Very Important Person* (VVIP). Namun pengawasan seperti ini memungkinkan terluputnya pengawasan pada tampilan CCTV oleh aparatur. Sedangkan CCTV hanya mampu merekam, namun tidak memiliki kemampuan *real-time situational awareness*. Kami merancang suatu sistem *line crossing detector*. *Line crossing detector* merupakan salah satu produk *real-time situational awareness* yang mampu memantau *key area* (area tertentu yang telah ditentukan sebagai area pengawasan/*clear area*). Sistem ini memiliki kemampuan untuk mendeteksi adanya objek yang melintas garis *virtual danger* dan atau *warning* yang telah ditetapkan, mengidentifikasi objek, dan mengirimkan informasi berupa notifikasi ketika terdapat objek tertentu yang melewati garis *virtual danger* dan atau *warning*. Desain implementasi untuk di dalam ruangan (*indoor area*). Secara khusus sistem ini dapat bekerja secara optimal pada tempat yang tidak padat aktivitas. *Line crossing detector* ini menggunakan *Spatial Sample Difference Consensus* (SSDC) untuk mendeteksi adanya objek yang bergerak. Kemudian dilakukan *tracking* objek menggunakan *centroid tracking*. Pengidentifikasi objek yang masuk ke dalam area pengawasan secara keseluruhan menggunakan model YOLO V3. Melalui metode *student t*, sistem mampu mendeteksi objek yang melewati garis pengawasan dengan tingkat kepercayaan 90% dengan tingkat signifikansi 10%.

**Kata Kunci:** *Line crossing detector, real-time situational awareness, spacial sample difference consensus, CCTV, pengawasan*

### **Abstract**

*Surveillance through Closed Circuit Television (CCTV) is one way to anticipate threats, both Very Important Persons (VIP) and or Very-Very Important Persons (VVIP). However, this kind of surveillance allows the apparatus to imprudent watch on the CCTV display. At the same time, CCTV only records the surveillance area, but it doesn't have real-time situational awareness ability. Line crossing detectors enable real-time situational awareness to monitor one of the key areas or some specific area determined as a monitoring area. This system can detect objects crossing the virtual danger and or warning lines that have been set, identify objects, and send information in the form of notifications when particular objects cross the virtual danger and or warning lines. The system that we built is specifically designed to work in the indoor area. This system can perform optimally in places that are not crowded with activities. It uses Spatial Sample Difference Consensus (SSDC) for detecting a moving object. In addition, Line Crossing Detector can process object tracking using centroid tracking and identify objects that enter the surveillance area using the YOLO V3 model. According to student t method, the system detects object crosses the line with 90% confidence and 10% significance.*

**Keywords:** *Line crossing detector, real-time situational awareness, spatial sample difference consensus, CCTV, surveillance*

## I. PENDAHULUAN

Salah satu pengamanan VIP (*Very Important Person*) dan VVIP (*Very-Very Important Person*) menggunakan *Closed Circuit Tele-Vision* (CCTV). CCTV ini ditempatkan pada tempat-tempat strategis. Kemudian pengawas mengawasi di pos pengamanan (*commander pos*) dengan zona akses terbatas (Peraturan Dirjen Perhubungan Udara Nomor KP 129 Tahun 2019 Tentang Petunjuk Teknik Pengawasan dan Investigasi Keamanan Penerbangan, 2017).

Walaupun dapat mengawasi secara *real-time*. CCTV hanya menampilkan *feed video* dari area yang diawasi (*clear area*). Ketika pengawas luput mengawasi tampilan CCTV (Ashby, 2017). Maka tidak ada yang CCTV lakukan, karena ia tidak memiliki kemampuan *real-time situational awareness*.

Kemampuan *real-time situational awareness* merupakan kemampuan dari suatu sistem untuk menyadari terjadinya suatu ancaman yang ada pada lingkungan di sekitarnya (Alkhomsan et al., 2017). Kemampuan ini menjadi sangat penting untuk membantu pengawas mengamankan VIP dan VVIP melalui CCTV.

Sistem *line crossing detector* ini merupakan bagian dari sistem *real-time situational awareness*, salah satu proyek milik PT LEN, Badan Usaha Milik Negara yang bergerak pada bidang elektronika pertahanan dan bagian dari *National Defence and High-tech Industries* (NDHI) dalam rangka pengembangan teknologi pertahanan dalam negeri.

Sistem ini bertujuan untuk membantu pengawas mengawasi suatu area melalui CCTV dengan penambahan kemampuan *real-time situational awareness*. Kemampuan ini diperoleh karena sistem mampu mengidentifikasi objek pada *clear area* dan apakah objek tersebut melewati penanda.

*Clear area* pada sistem tidak ditandai secara khusus oleh pengawas pada monitor CCTV. Area ini ditandai pada jalur masuknya yaitu dengan *virtual danger* dan *virtual warning*.

Kemampuan identifikasi objek-objek yang masuk ke dalam area pengawasan menggunakan model YOLO (*You only Look Once*), sedangkan kemampuan mengetahui objek melewati penanda menggunakan *line crossing detector*. *Line crossing detector* biasa digunakan untuk mengawasi suatu area dan merekam pergerakan objek yang masuk ataupun keluar dari area pengawasan tersebut (Ahmed et al., 2021).

Untuk mengetahui apakah objek melewati garis *virtual danger* dan atau *warning* menggunakan

teknik *object tracking*, pada sistem menggunakan *centroid tracking algorithm*.

Ada berbagai macam metode untuk mendeteksi suatu objek yang bergerak melalui sebuah citra, dimulai dari *background subtraction* (Jusman et al., 2020; Zhou & Maskell, 2017), *frame difference* (Jusman et al., 2020; Nakashima & Yabuta, 2018), *temporal differencing* (Boufares et al., 2021), dan *optical flow* (J. Zhang et al., 2019). Setiap algoritma memiliki kelebihan dan kekurangannya masing-masing.

*Spatial Sample Difference Consensus* (SSDC) digunakan pada penelitian ini untuk deteksi objek bergerak. SSDC unggul dibandingkan dengan algoritma *background subtraction* lainnya seperti ViBE dan *Gaussian Mixture Model* GMM (Z. Zhang et al., 2018). Algoritma SSDC menghasilkan derau citra *background* yang lebih sedikit dibandingkan dengan dua algoritma lainnya.

Pada penelitian ini secara khusus kemampuan mengenali objek yang melewati garis *virtual danger* dan garis *virtual warning* dianalisis tingkat keberhasilannya menggunakan metode uji *student t*.

## II. METODE

Penelitian ini terdiri dari beberapa tahapan. Tahapan pertama memahami proses bisnis. Untuk itu dilakukan studi pustaka dan analisis sistem, seperti analisis sistem pengawasan CCTV, kebutuhan data dan model, dan kebutuhan sistem.

Perancangan sistem masuk tahap kedua. Pada tahap ini dirancang arsitektur sistem, fisik sistem, dan logika sistem.

Terakhir tahapan ketiga, implementasi. Pengembangan perangkat lunak dilakukan dengan kombinasi *scrum framework* untuk pengelolaan tim. *Incremental software development life cycle* sebagai pendekatan pembangunan sistem.

Selanjutnya tahapan terakhir adalah pengujian pada kemampuan utama sistem.

### Proses Bisnis

Setelah melakukan studi pustaka, analisis terhadap sistem CCTV, dan penyesuaian dengan kebutuhan PT. LEN. Diperoleh *Functional requirements* seperti Tabel 1.

Tabel 1. *Functional Requirements*

Kode	Keterangan
FR-U-1-001	Sistem dapat menerima <i>feed camera</i> dengan resolusi 1080P @60fps.
FR-U-1-002	Sistem dapat menampilkan <i>feed camera</i> dengan resolusi 1080P @30fps.

FR-U-2-001	Pengawas dapat menggambar garis virtual <i>danger</i> pada sistem.	FR-U-15-001	Pengawas dapat mengurungkan garis virtual <i>danger</i> yang terakhir digambar pada <i>frame</i> .
FR-U-2-002	Sistem dapat menampilkan pilihan arah masuk pada garis virtual <i>danger</i> .	FR-U-16-001	Sistem dapat menampilkan peringatan ketika tidak terdapat garis virtual yang digambar pada sistem.
FR-U-2-003	Pengawas dapat memilih arah masuk pada garis virtual <i>danger</i> .		
FR-U-3-001	Pengawas dapat menggambar garis virtual <i>warning</i> pada sistem.		
FR-U-3-002	Sistem dapat menampilkan pilihan arah masuk pada garis virtual <i>warning</i> .		
FR-U-3-003	Pengawas dapat memilih arah masuk pada garis virtual <i>warning</i> .		
FR-U-4-001	Sistem mem- <i>bounding box</i> pada objek bergerak yang terdeteksi.		
FR-U-4-002	Sistem mendeteksi objek bergerak.		
FR-U-5-001	Sistem mendeteksi adanya objek yang melewati garis virtual <i>danger</i> .		
FR-U-6-001	Sistem dapat menentukan arah masuk suatu objek yang telah melewati garis virtual <i>danger</i> .		
FR-U-7-001	Sistem dapat mengidentifikasi objek yang melewati garis virtual <i>danger</i> .		
FR-U-8-001	Sistem dapat memberikan notifikasi jika objek melewati garis <i>danger</i> . Informasi dimuat pada notifikasi adalah “terdapat objek masuk”, dan kategori dari objek berupa hewan, manusia, atau kendaraan.		
FR-U-8-002	Sistem dapat menyimpan <i>log</i> berisi data mengenai jenis garis virtual yang dilewati, <i>class object</i> melewati garis virtual <i>danger</i> dan waktu sistem menangkap objek yang melewati garis virtual <i>danger</i> .		
FR-U-9-001	Sistem dapat mendeteksi adanya objek yang melewati garis virtual <i>warning</i> .		
FR-U-10-001	Sistem dapat menentukan arah masuk objek yang melewati garis virtual <i>warning</i> .		
FR-U-11-001	Sistem dapat mengidentifikasi objek yang telah melewati garis virtual <i>warning</i> .		
FR-U-12-001	Sistem dapat memberikan notifikasi objek yang melewati garis virtual <i>warning</i> . Informasi adalah keterangan “terdapat objek masuk”, dan kategori dari objek.		
FR-U-12-002	Sistem dapat menyimpan <i>log</i> berisi data mengenai jenis garis virtual yang dilewati, <i>class object</i> yang masuk melewati garis virtual <i>warning</i> dan waktu sistem menangkap objek yang melewati garis virtual <i>warning</i> .		
FR-U-13-001	Sistem dapat menampilkan peringatan ketika tidak ada perangkat kamera pengawas yang tersambung dengan sistem.		
FR-U-14-001	Pengawas dapat mengurungkan garis virtual <i>warning</i> yang terakhir digambar pada <i>frame</i> .		

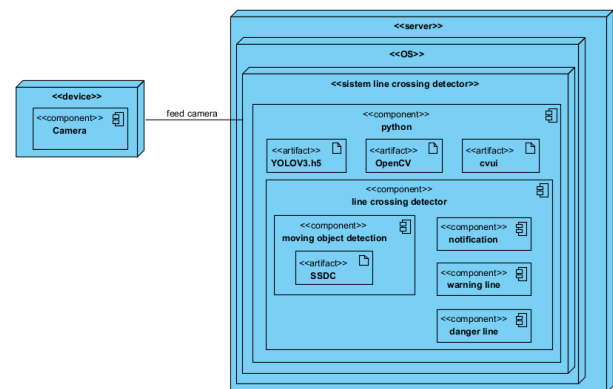
**Perancangan**

Bagian perancangan dirancang arsitektur sistem, interaksi dengan lingkungan, dan logika sistem.

**Perancangan Arsitektur Sistem**

Arsitektur sistem terbagi tiga *layer* terdiri dari sensor, *processing*, dan *end-user*. *Layer* sensor merepresentasikan alat untuk akuisisi citra, *layer processing* merepresentasikan alat memproses hasil citra setelah diakuisisi.

Selanjutnya *layer end-user*, sistem dibangun *platform desktop*. Rancangan *deployment* dapat dilihat pada Gambar 1.



Gambar 1. Rancangan *Deployment* Sistem *Line Crossing Detector* Terdiri Dari Tiga *Layer*

Pada Gambar 1, server digunakan untuk proses pengujian sistem. Pada server ter-*install* OS (*Operating System*), sistem berjalan pada OS ini, dan mendapat *feed* dari kamera. Komponen yang digunakan untuk mendukung sistem antara lain YOLOv3, OpenCV, dan cvui.

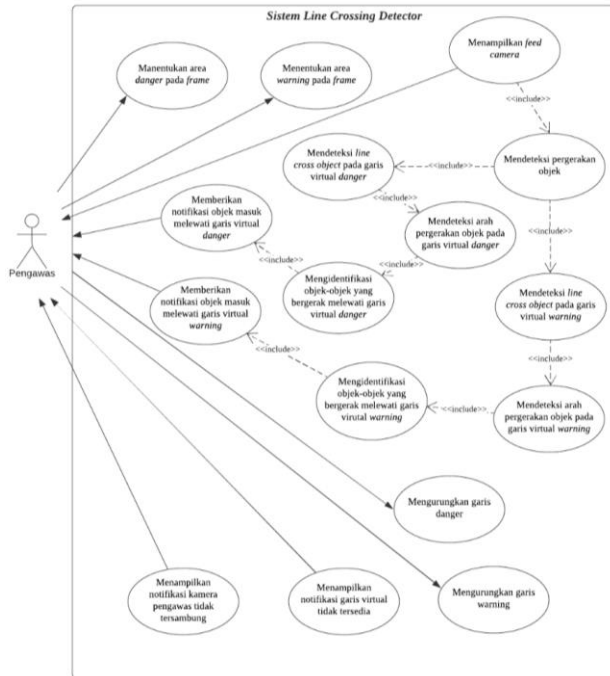
Kemampuan aplikasi untuk mengetahui objek bergerak menggunakan algoritma SSOC. Kemampuan menandai tampilan CCTV seperti menandai *warning line* dan *danger line*. Penandaan ini memicu *notification* dari sistem kepada pengguna.

**Perancangan Interaksi Sistem**

Interaksi sistem dengan lingkungan eksternalnya digambarkan menggunakan diagram UML. Lingkungan eksternal yang dimaksud berupa aktor, maupun sistem lain yang berhubungan

secara langsung dengan sistem. Rancangan dapat dilihat pada Gambar 2.

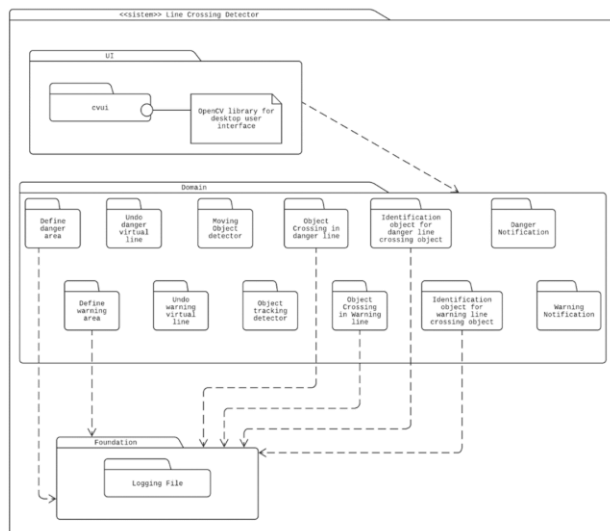
Terdapat enam belas *use case* menunjukkan sistem secara langsung berhubungan dengan satu aktor yakni pengawas. Pengawas berperan sebagai *primary actor*.



Gambar 2. Use Case Interaksi Sistem Dengan Aktor

**Perancangan Logika Sistem**

Rancangan logika sistem akan ditampilkan menggunakan *package diagram* seperti pada Gambar 3. Pada Gambar 3 dapat dilihat logika sistem terdiri dari tiga *layer*.



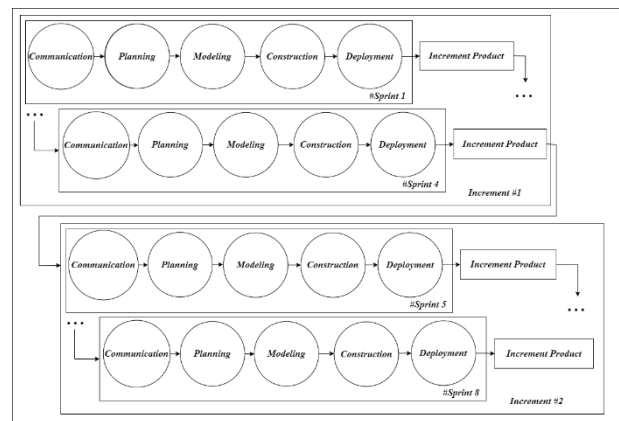
Gambar 3. Rancangan logika sistem

*Layer* pertama merepresentasikan *UI layer*, menggunakan *library cvui* untuk tampilan *desktop*. *Domain layer* merepresentasikan proses bisnis (*core application* yang ditangani oleh sistem). Terdapat dua belas proses bisnis yang ditangani oleh sistem. *Layer* terakhir, *foundation layer*. *Layer foundation* merepresentasikan *core service* secara *low-level technical service* yang menunjukkan kemampuan sistem secara teknis.

Secara umum sistem memiliki kemampuan untuk mencatat *logging system*. Kemampuan ini digunakan untuk menyimpan *data crossing object* yang masuk melewati garis *virtual danger* dan atau *warning*. Secara teknis, *low-level technical service* merupakan *service* dapat diakses oleh pengguna.

**Implementasi**

Proses pembangunan sistem menggunakan *scrum framework* sebagai *framework* untuk pengelolaan tim dan *incremental Software development life cycle (SDLC)* sebagai pendekatan pembangunan sistem. *SDLC incremental* merupakan model yang mengintegrasikan alur linear dengan *iterative* sehingga pada proses akhirnya menghasilkan sebuah inkremental produk (Pressman & Maxim, 2015).



Gambar 4. Proses Inkremental pada Sistem

Gambar 4 merupakan ilustrasi kombinasi antara *scrum framework* dengan *SDCL* inkremental. Pembangunan sistem dilaksanakan dalam dua *increment* dan empat *sprint* untuk setiap *increment*. Setiap *sprint* terdiri dari lima tahapan, yaitu *communication*, *planning*, *modelling*, *construction*, dan *deployment*. *Increment product* merupakan hasil implementasi berupa sistem dari *sprint* yang baru final dengan *sprint* periode sebelumnya.

**Pengujian**

Jenis pengujian yang dilakukan adalah *functional testing*, *regression testing*, dan *user acceptance testing*.

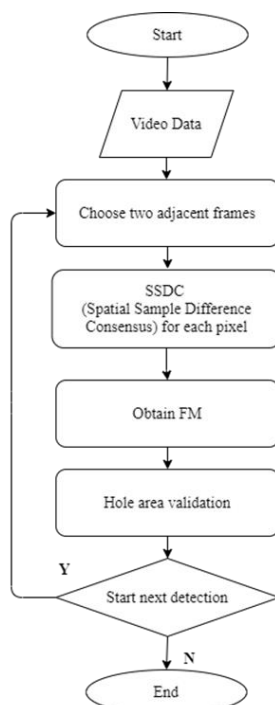
*Functional testing*, dilakukan fitur sistem. *Regression testing*, dilakukan untuk proses integrasi fungsi CCTV yang telah ada dengan fungsi yang baru. Tujuannya melihat apakah sistem masih berjalan dengan baik setelah integrasi. Kemudian *user acceptance test* untuk memastikan bahwa sistem sesuai dengan harapan *product owner* (PT. LEN) ketika fase *increment* selesai.

### III. HASIL DAN PEMBAHASAN

Secara garis besar sistem memiliki bagian pengolahan citra digital dengan komponen berdasarkan algoritma SSDC, *centroid tracking* untuk *object tracking*, dan deteksi objek dengan YOLOv3.

#### SSDC

Pengolahan citra digital merupakan sebuah bentuk pengolahan atau pemrosesan informasi berupa gambar yang akan ditransformasikan menjadi sebuah gambar baru sebagai keluarannya dengan teknik tertentu (Peters, 2017). Pengolahan citra digital digunakan untuk menjalankan algoritma SSDC.



Gambar 5. Alur SSDC

SSDC merupakan sebuah perhitungan matematika untuk membandingkan dua *sample frame* yaitu  $n$  dan  $n + 1$  dalam  $N$  jumlah sampel spasial untuk setiap piksel di setiap *frame*. *Frame-frame* tersebut dianalisis kembali untuk memastikan kumpulan *pixel p* dalam  $n - x_n(p)$  yang telah diekstrak merupakan objek bergerak

atau bukan dengan menentukan *background pixel* dan *foreground pixel* (C. Zhang et al., 2017).

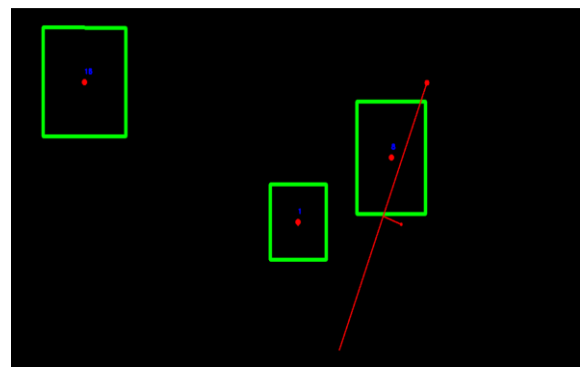
Gambar 5 diambil dari (C. Zhang et al., 2017), menyatakan SSDC memiliki tiga fase esensial yaitu memilih dua *frame* (*choose two adjacent frames*), mendapatkan objek bergerak (*Obtain Foreground Mask*), dan memvalidasi area (*Hole area validation*).

Tantangan terbesar dalam mendeteksi objek bergerak dalam suatu *frame* adalah bagaimana menghilangkan gangguan atau derau seperti goyangan kamera, perubahan cahaya, dan perubahan lingkungan (Zeyi & Gang, 2021).

#### Centroid Tracking

Setelah mendeteksi objek, selanjutnya *object* di-track pada video. Cara ini biasa digunakan untuk menjadi solusi untuk mendeteksi pergerakan orang (Challa et al., 2011).

*Tracking* dilakukan dengan membandingkan *object centroid* pada objek yang telah dilacak oleh *centroid tracker* dengan *centroid object frame* berikutnya dalam video. Algoritma ini mengasumsikan objek terdeteksi di setiap frame dan ditandai *bounding box* yang memiliki titik koordinat (x,y). Setiap *bounding box* masing-masing memiliki id yang bersifat unik (Bakliwal et al., 2020).



Gambar 6. Ilustrasi *Centroid Tracking*

Titik tengah *bounding box* (*centroid*) pada Gambar 6 berperan dalam proses *line crossing*. *Centroid* suatu objek dengan *centroid* objek tersebut pada *frame* selanjutnya membentuk suatu vektor. Perpotongan vektor ini dengan garis virtual menunjukkan objek melintas. Arah vektor merupakan arah pergerakan objek.

#### Pendeteksian Objek

Pendeteksian objek menggunakan YOLOv3. YOLO dipilih karena akurasi deteksi objeknya dapat mencapai 90% (Manajang et al., 2020). Model YOLO terbukti *powerful*, contoh: mampu mendeteksi komponen elektronik pada *printed*

circuit board (PCB) (Li et al., 2019), pengawasan kemacetan (Masurekar et al., 2020), *monitoring social distancing* dalam situasi Covid-19 (Punn et al., 2020), dan mendeteksi hama tanaman tomat (Liu & Wang, 2020).

YOLO V3 dilatih dengan COCO dataset. Terdapat 80 objek yang dapat diidentifikasi dengan YOLO V3 (Mike, 2018). Namun pada implementasi penelitian ini, proses identifikasi hanya mengidentifikasi tiga *super class* dapat dilihat pada Tabel .

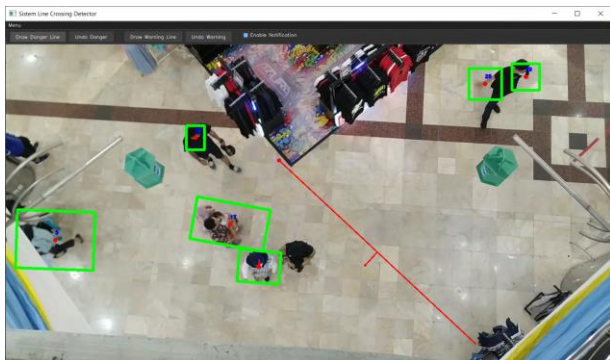
Tabel 2. Objek Pemicu Notifikasi Serta Hubungan dengan Garis *Danger* dan atau *Warning*

Category Object	Super class object	Pemicu garis danger	Pemicu garis warning
Person	person	Ya	Ya
Cat	animal	Ya	Ya
Motorcycle	vehicle	Tidak	Ya

*Super class* ini dipilih karena hasil analisis risiko yang dimiliki berdasarkan penelitian mengenai *surveillance* untuk meningkatkan keamanan pada ruang publik (Socha & Kogut, 2020). Alasan lain hewan atau *animal* dapat terlibat dalam aktivitas kriminal, baik oleh hewan itu sendiri, maupun sebagai partner manusia (Beirne, 1995).

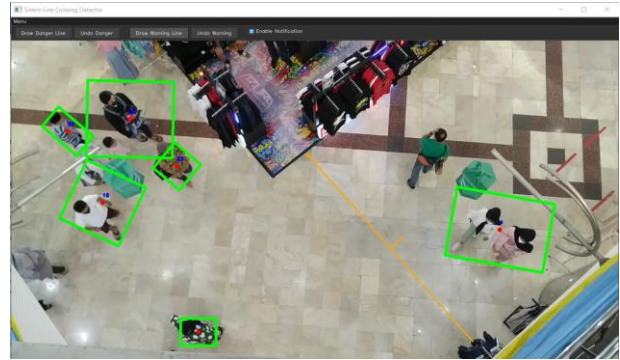
### Hasil Implementasi

Gambar 7 merupakan hasil implementasi untuk menggambar garis *virtual danger*. Pada Gambar 7 dapat dilihat ada garis berwarna merah. Garis itu muncul setelah digambar oleh pengawas. Setelah garis muncul pengawas harus menentukan arah yang dinyatakan masuk. *Bounding box* hijau merupakan objek *super class* yang terdeteksi.



Gambar 7. Implementasi Garis *Virtual Danger*

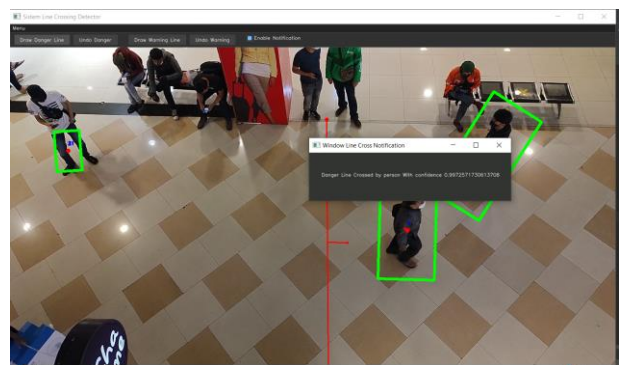
Gambar 8 merupakan hasil implementasi untuk menggambar garis *virtual warning*.



Gambar 8. Implementasi Garis *Virtual Warning*

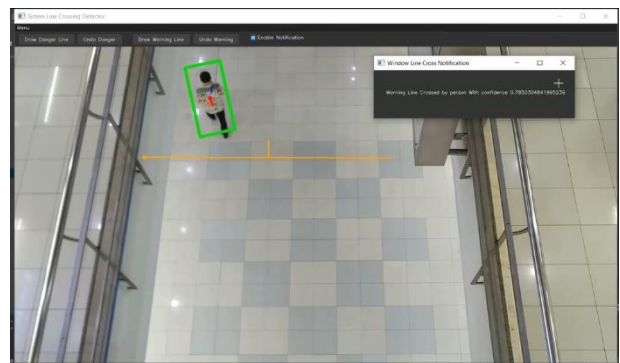
Pada Gambar 8 ada garis berwarna jingga, dan ada garis penunjuk arah area yang dilindungi. Objek yang bergerak keluar dari area tersebut tidak akan menghasilkan notifikasi pada sistem.

Hasil implementasi dari notifikasi ketika terdapat suatu objek yang masuk ke dalam area pengawasan melewati garis *virtual danger* dapat dilihat pada Gambar 9.



Gambar 9. Implementasi Notifikasi Sistem Ketika Objek Bergerak Masuk Area Ditandai Garis *Virtual Danger*

Gambar 10 merupakan hasil implementasi dari notifikasi ketika suatu objek masuk ke dalam area pengawasan melewati garis *virtual warning*.



Gambar 10. Implementasi Notifikasi Sistem Ketika Objek Bergerak Masuk Area Ditandai Garis *Virtual Warning*

**Pengujian Objek Melewati Garis *Virtual Danger***

Pengujian objek melewati garis virtual *danger* menggunakan 13 skenario, dengan 106 kali percobaan dengan menguji seluruh skenario untuk setiap video pengujian. Tabel merupakan hasil pengujian. Dengan Perumusan hipotesis awal:

**H<sub>0</sub>**:  $\mu =$  sistem *line crossing detector* tidak memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *danger*.

**H<sub>a</sub>**:  $\mu =$  sistem *line crossing detector* memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *danger*.

Tabel 3. Hasil Percobaan *Line Crossing Danger*

Kode test Skenario	Total Akurasi (%)
TCL001	100
TCL002	90,909
TCL003	94,736
TCL004	100
TCL005	100
TCL006	94,117
TCL013	100
TCL014	50
TCL015	75
TCL016	100
TCL017	100
TCL018	100
TCL025	100
<b>Rata-rata</b>	<b>92,674</b>

Perhitungan dari t hitung terhadap hasil pengujian objek melewati garis virtual *danger*.

Mencari t hitung dari percobaan

Diketahui :

$$\bar{x} = 92,674 \quad s = 14,6409$$

$$\mu_0 = 90 \quad n = 106$$

Maka t hitung:

$$t = \frac{92,674 - 90}{14,6409 / \sqrt{106}} = 1,88047$$

Mencari nilai df atau derajat kebebasan dengan menggunakan persamaan

$$df = n - 1$$

Diketahui :

$$n = 106$$

Maka df :

$$df = 106 - 1 = 105$$

Menentukan t<sub>tabel</sub>

Diketahui:

$$\alpha = 0.1 \quad \frac{\alpha}{2} = 0.05$$

Nilai dari t<sub>tabel</sub> pada kasus untuk uji *student t* objek melewati garis virtual *danger* ini dengan

melihat tabel t, nilai t diambil dari nilai df dan  $\frac{\alpha}{2}$  yang telah didapatkan sebelumnya.

$$t_{tabel} = 1.65950$$

Menyimpulkan hipotesis pada  $\alpha = 0.1$ :

Diketahui :

$$t_{hitung} = 1,88047 \quad t_{tabel} = 1,65950$$

Maka:

Sisi kanan:  $t_{hitung} = 1,88047 > t_{(0.005, 105)} = 1.65950$  (**H<sub>0</sub> ditolak**)

Sisi kiri:  $-t_{hitung} = -1,88047 < -t_{(0.005, 105)} = -1.65950$  (**H<sub>0</sub> ditolak**)

**H<sub>0</sub> ditolak**, kesimpulan dari hasil uji statistik, sistem *line crossing detection* dapat mendeteksi objek yang melewati garis virtual *danger* dapat mencapai akurasi 90% untuk pendeteksian objek yang melewati garis virtual *danger*.

**Pengujian Objek Melewati Garis *Virtual Warning***

Pengujian bagian ini juga menggunakan tiga belas skenario, dengan 100 percobaan. Tabel merupakan tabel hasil pengujian dari *object crossing* pada garis virtual *danger*. Dengan Perumusan hipotesis awal:

**H<sub>0</sub>**:  $\mu =$  sistem *line crossing detector* tidak memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *warning*.

**H<sub>a</sub>**:  $\mu =$  sistem *line crossing detector* memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *warning*.

Tabel 4. Hasil Pengujian *Line Crossing Warning*

Kode test Skenario	Total Akurasi (%)
TCL007	100,000
TCL008	91,667
TCL009	94,118
TCL010	100,000
TCL011	100,000
TCL012	90,909
TCL019	100,000
TCL020	66,667
TCL021	60,000
TCL022	100,000
TCL023	100,000
TCL024	100,000
TCL026	100,000
<b>Rata-rata</b>	<b>92,566</b>

Berikut merupakan perhitungan dari t hitung terhadap hasil pengujian objek melewati garis virtual *warning*.

Mencari t hitung dari percobaan

Diketahui :

$$\bar{x} = 92,024 \quad s = 13,4765$$

$$\mu_0 = 90 \quad n = 100$$

Maka t hitung:

$$t = \frac{92,024 - 90}{13,4765 / \sqrt{100}} = 1,90417$$

Mencari nilai df atau derajat kebebasan dengan menggunakan persamaan

$$df = n - 1$$

Diketahui :

$$n = 100$$

Maka df :  $df = 100 - 1 = 99$

Menentukan t tabel

Diketahui:

$$\alpha = 0.1 \quad \alpha/2 = 0.05$$

Nilai dari t tabel pada kasus untuk uji *student t* objek melewati garis virtual *warning* ini dengan melihat tabel t, nilai t diambil dari nilai df dan  $\alpha/2$  yang telah didapatkan sebelumnya.

$$t_{(0.005, 99)} = 1.66039$$

Menyimpulkan hipotesis

Diketahui :

$$t_{hitung} = 1,90417 \quad t_{tabel} = 1.66039$$

Maka:

Sisi kanan:  $t_{hitung} = 1,90417 > t_{(0.005, 99)} = 1.660397$   
( $H_0$  ditolak)

Sisi kiri:  $-t_{hitung} = -1,90417 < -t_{(0.005, 99)} = -1.66039$   
( $H_0$  ditolak)

**$H_0$  ditolak**, kesimpulan, sistem *line crossing detection* dapat mendeteksi objek yang melewati garis virtual *warning* dapat mencapai akurasi 90% untuk pendeteksian objek yang melewati garis virtual *warning*.

### Pengujian Objek Melewati Garis Virtual *Danger* dan *Warning*

Pengujian ini menggunakan empat skenario, dengan 98 kali percobaan. Tabel 5 hasil pengujian dari *object crossing* pada garis virtual *danger* dan *warning*. Perumusan hipotesis awal:

**$H_0$** :  $\mu =$  sistem *line crossing detector* tidak memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *danger* dan *warning*.

**$H_a$** :  $\mu =$  sistem *line crossing detector* memiliki akurasi lebih dari atau sama dengan 90% untuk mendeteksi adanya *object crossing* pada garis virtual *danger* dan *warning*.

Tabel 5. Hasil Pengujian *Line Crossing Danger* dan *Line Crossing Warning*

Kode test Skenario	Rata-rata Akurasi (%)
TCL027	88,889
TCL028	88,889
TCL029	100
TCL030	100
<b>Rata-rata</b>	<b>94,4445</b>

Berikut merupakan perhitungan dari t hitung terhadap hasil pengujian objek melewati garis virtual *danger* dan *warning*.

Mencari t hitung dari percobaan

Diketahui :

$$\bar{x} = 94,4445 \quad s = 0$$

$$\mu_0 = 90 \quad n = 98$$

Maka t hitung:

$$t = \frac{94,4445 - 90}{6,4149 / \sqrt{18}} = 2,93945$$

Mencari nilai df atau derajat kebebasan dengan menggunakan persamaan

$$df = n - 1$$

Diketahui :

$$n = 18$$

Maka df :

$$df = 18 - 1 = 17$$

Menentukan t tabel

Diketahui:

$$\alpha = 0.1 \quad \alpha/2 = 0.05$$

Nilai dari t tabel pada kasus untuk uji *student t* objek melewati garis virtual *danger* dan *warning* ini dengan melihat tabel t, nilai t diambil dari nilai df dan yang telah didapatkan sebelumnya.

Maka :

$$t_{(0.005, 17)} = 1.73961$$

Menyimpulkan hipotesis

Diketahui :

$$t_{hitung} = 2,93945 \quad t_{tabel} = 1.73961$$

Maka:

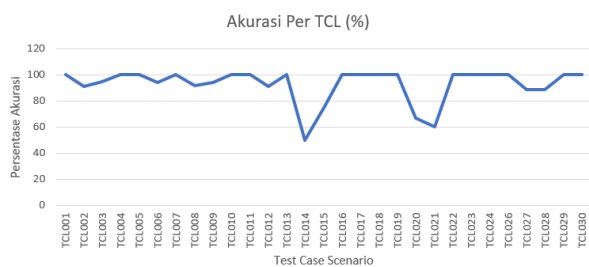
Sisi kanan:  $t_{hitung} = 1.73961 > t_{(0.005, 17)} = 1.73961$   
( $H_0$  ditolak)

Sisi kiri:  $-t_{hitung} = -1.73961 < -t_{(0.005, 17)} = -1.73961$   
( $H_0$  ditolak)

**$H_0$  ditolak**, kesimpulannya, sistem *line crossing detection* dapat mendeteksi objek yang melewati garis virtual *danger* dan *warning* secara bersamaan dapat mencapai akurasi 90% untuk pendeteksian objek yang melewati garis virtual.

Gambar 11 menunjukkan akurasi sistem *line crossing detector* yang telah dibangun dalam mendeteksi objek bergerak melintasi garis virtual berdasarkan tiga puluh skenario pengujian.





Gambar 11. Persentase Akurasi Deteksi Objek yang Melintasi Garis Virtual

#### IV. PENUTUP

##### Kesimpulan

Sistem *line crossing detector* dengan menggunakan metode algoritma *spatial sample difference consensus*, *centroid*, dan *COCO dataset* telah berhasil dibangun dengan karakteristik sebagai berikut: (1) sistem mampu mendeteksi objek ketika melintasi garis virtual *danger* dan atau *warning* yang telah ditetapkan, dengan tingkat kepercayaan 90% dan tingkat signifikansi 10%, (2) sistem mampu mengidentifikasi objek setelah melintasi garis virtual *danger* dan atau *warning* yang telah ditetapkan, (3) sistem mampu memberikan notifikasi ketika terdapat objek melewati garis virtual *danger* dan atau *warning* yang telah ditetapkan.

##### Saran

Saran untuk pengembangan sistem antara lain: (1) Diperlukan kemampuan untuk menghitung kecepatan objek bergerak tertangkap kamera pengawas, (2) pengoptimalan sistem dapat dilakukan dengan implementasi *multithread* atau *microservice*, (3) tingkatkan kemampuan *tracking* untuk video yang tidak stabil.

##### Ucapan Terima Kasih

Karya ini didukung penuh oleh Politeknik Negeri Bandung.

#### V. DAFTAR PUSTAKA

- Ahmed, I., Ahmad, M., Ahmad, A., & Jeon, G. (2021). IoT-based crowd monitoring system: Using SSD with transfer learning. *Computers and Electrical Engineering*, 93. <https://doi.org/10.1016/j.compeleceng.2021.107226>
- Alkhomsan, M. N., Hossain, M. A., Rahman, S. M. M., & Masud, M. (2017). Situation Awareness in Ambient Assisted Living for

Smart Healthcare. *IEEE Access*, 5, 20716–20725.

<https://doi.org/10.1109/ACCESS.2017.2731363>

- Ashby, M. P. J. (2017). The Value of CCTV Surveillance Cameras as an Investigative Tool: An Empirical Analysis. *European Journal on Criminal Policy and Research*, 23(3), 441–459. <https://doi.org/10.1007/s10610-017-9341-6>
- Bakliwal, A., Puranik, A., Modi, A., Jain, A., Jaiswal, A., Godani, D., Bhanodia, P., & Jangde, P. (2020). *Crowd Counter: an Application of Centroid Tracking Algorithm*. 04, 1138–1141.
- Bierne, P. (1995). The Use and Abuse of Animals in Criminology: A Brief History and Current Review. *Social Justice*, 22(1), 5–31.
- Boufares, O., Boussif, M., & Aloui, N. (2021). Moving Object Detection System Based on the Modified Temporal Difference and OTSU algorithm. *18th IEEE International Multi-Conference on Systems, Signals and Devices, SSD 2021*, 1378–1382. <https://doi.org/10.1109/SSD52085.2021.9429516>
- Challa, S., Morelande, M. R., Mušicki, D., & Evans, R. J. (2011). Fundamentals of object tracking. In *Fundamentals of Object Tracking* (1st ed., Vol. 9780521876). Cambridge University Press. <https://doi.org/10.1017/CBO9780511975837>
- Jusman, Y., Hinggis, L., Wiyagi, R. O., Isa, N. A. M., & Mujaahid, F. (2020). Comparison of Background Subtraction and Frame Differencing Methods for Indoor Moving Object Detection. *Proceeding - 1st International Conference on Information Technology, Advanced Mechanical and Electrical Engineering, ICITAMEE 2020*, 214–219. <https://doi.org/10.1109/ICITAMEE50454.2020.9398484>
- Li, J., Gu, J., Huang, Z., & Wen, J. (2019). Application research of improved YOLO V3 algorithm in PCB electronic component detection. *Applied Sciences (Switzerland)*, 9(18). <https://doi.org/10.3390/app9183750>
- Liu, J., & Wang, X. (2020). Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network. *Frontiers in Plant Science*, 11. <https://doi.org/10.3389/fpls.2020.00898>
- Manajang, D., Dompie, S., & Jacobus, A. (2020). Implementasi Framework Tensorflow Object

- Detection Dalam Mengklasifikasi Jenis Kendaraan Bermotor. *Jurnal Teknik Informatika*, 15(3), 171–178.
- Measurekar, O., Jadhav, O., Kulkarni, P., & Patil, S. (2020). Real Time Object Detection Using YOLOv3. *International Research Journal of Engineering and Technology (IRJET)*, 07(03), 3764–3768. [www.irjet.net](http://www.irjet.net)
- Mike. (2018). *What Object Categories/Labels Are In COCO Dataset?* <https://tech.amikelive.com/node-718/what-object-categories-labels-are-in-coco-dataset/>
- Nakashima, T., & Yabuta, Y. (2018). Object Detection by using Interframe Difference Algorithm. *Proceedings - 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Mechatronics 2018*, 98–102. <https://doi.org/10.1109/MECATRONICS.2018.8495743>
- Peters, J. F. (2017). *Foundations of Computer Vision* (J. Kacprzyk & L. C. Jain (eds.); 1st ed.). Springer International Publishing Switzerland. <https://doi.org/10.1007/978-3-319-52483-2>
- Pressman, R. S., & Maxim, B. R. (2015). Software engineering: A practitioner's approach. In *Advances in Engineering Software (1978)* (8th ed., Vol. 5, Issue 3). Mc Graw Hill.
- Punn, N. S., Sonbhadra, S. K., Agarwal, S., & Rai, G. (2020). *Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques*. <http://arxiv.org/abs/2005.01385>
- Peraturan Direktur Jenderal Perhubungan Udara Nomor KP 129 Tahun 2019 Tentang Petunjuk Teknik Pengawasan dan Investigasi Keamanan Penerbangan, 516 (2017).
- Socha, R., & Kogut, B. (2020). Urban video surveillance as a tool to improve security in public spaces. *Sustainability (Switzerland)*, 12(15). <https://doi.org/10.3390/su12156210>
- Zeyi, Z., & Gang, L. (2021). Target motion detection algorithm based on dynamic threshold. *Journal of Physics: Conference Series*, 1738(1). <https://doi.org/10.1088/1742-6596/1738/1/012085>
- Zhang, C., Zheng, J., Zhang, Y., Han, M., & Li, B. (2017). Moving object detection algorithm based on pixel spatial sample difference consensus. *Multimedia Tools and Applications*, 76(21), 22077–22093. <https://doi.org/10.1007/s11042-017-4802-y>
- Zhang, J., Ding, Y., Xu, H., & Yuan, Y. (2019). An optical flow based moving objects detection algorithm for the UAV. *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019*, 233–238. <https://doi.org/10.1109/CCOMS.2019.8821661>
- Zhang, Z., Trivedi, C., & Liu, X. (2018). Automated detection of grade-crossing-trespassing near misses based on computer vision analysis of surveillance video data. *Safety Science*, 110, 276–285. <https://doi.org/10.1016/j.ssci.2017.11.023>
- Zhou, Y., & Maskell, S. (2017). Moving object detection using background subtraction for a moving camera with pronounced parallax. *2017 Symposium on Sensor Data Fusion: Trends, Solutions, Applications, SDF 2017, 2017-Decem*, 1–6. <https://doi.org/10.1109/SDF.2017.8126361>