

Monitoring Ruang Untuk Deteksi Manusia Berbasis CNN Dengan Fitur Push Notification

Windra Swastika
Ma Chung Human Machine
Interaction Research Center
Universitas Ma Chung
windra.swastika@machung.ac.id

Albert Wahyudi Nur
Program Studi Teknik Informatika
Universitas Ma Chung
albertwnur@gmail.com

Oesman Hendra Kelana
Program Studi Teknik Informatika
Universitas Ma Chung
oesman.hendra@machung.ac.id

Abstrak – Pemantauan atau *monitoring* adalah salah satu kegiatan untuk memantau setiap gerakan dari waktu ke waktu. Pemantauan memiliki berbagai tujuan termasuk menganalisis, mengumpulkan data atau mengamati pola pergerakan. Pemantauan juga dapat diartikan, yaitu melakukan pengamatan terhadap situasi di sekitarnya. Dalam penelitian ini, dibuat sebuah sistem *monitoring* ruang yang dapat mendeteksi objek manusia dengan menggunakan perangkat Raspberry Pi 3 dan akan mengirimkan *push notification* ketika sistem mendeteksi gerakan manusia dalam video. Untuk deteksi manusia berbasis *Convolutional Neural Network* (CNN) digunakan *framework You Only Look Once* (YOLO) dan untuk mengirim *notification* digunakan teknologi *Firebase Cloud Messaging*. Dari pengujian yang dilakukan, didapatkan akurasi pendeteksian sebesar 71% untuk video pertama dan akurasi 73,2% untuk video kedua. Sensitivitas seluruh sistem adalah 55,01% dan 57,96% serta spesifisitas dari kedua tes ini adalah 95% dan 94,77%

Kata Kunci: *Convolutional Neural Network*, Deteksi Gerakan, *Monitoring Ruang*, Raspberry Pi, YOLO

I. PENDAHULUAN

Kasus pencurian yang terjadi di kawasan tempat tinggal atau perumahan merupakan salah satu bentuk kejahatan pencurian yang banyak mendapat perhatian. Kejahatan yang umumnya terjadi di lingkungan tempat tinggal berbentuk pencurian rumah kosong, pencurian kendaraan bermotor, pencurian dengan penipuan, perampokan, serta pembobolan rumah (*breaking & entering*) [1]. Upaya pencegahan terhadap terjadinya kasus pencurian bisa dilakukan oleh calon korban maupun upaya dari pihak kepolisian. Berdasarkan Undang-Undang No.2 Tahun 2002 Pasal 13 tentang Tugas Pokok Kepolisian, disebutkan bahwa pihak kepolisian telah melakukan tindakan pencegahan (*preventive*), pendekatan (*preemptive*), dan penegakan hukum (*repressive*). Tindakan lain yang dapat dilakukan adalah dengan *monitoring* rumah atau ruangan, sehingga pemilik rumah dapat memantau atau mengawasi rumah atau ruangan dari jarak jauh. *Monitoring* ruang membutuhkan bantuan alat. Adapun alat yang paling banyak digunakan adalah *Closed Circuit Television* (CCTV) dan *IP Camera*.

Beberapa kekurangan dari kedua alat tersebut adalah kebutuhan instalasi terlebih dahulu; memerlukan kabel, konektor, adaptor, dan material bantu lainnya saat instalasi (penambahan biaya); lebih mudah dipengaruhi interferensi dan bisa disadap; rekaman gerakan umumnya mengalami perlambatan atau *delay* bergantung pada panjang kabel (CCTV) atau spesifikasi kamera pada *IP Camera*. Hal tersebut menjadi hambatan serius apabila pengguna membutuhkan solusi *monitoring* dengan biaya lebih terjangkau, praktis dan instalasi mudah, serta membutuhkan sedikit daya.

Hingga saat ini, telah diciptakan beberapa perangkat *monitoring* ruang atau keadaan sekitar dengan memanfaatkan perangkat IoT dan *microcontroller*. Perangkat tersebut antara lain ATmega328, Arduino 101, dan Raspberry Pi. Metode pendukung yang digunakan untuk melakukan *monitoring* juga bermacam-macam. Metode yang digunakan adalah dengan menggunakan sensor *Passive Infra Red* (PIR), lalu ada juga dengan cara mendeteksi panas yang dikeluarkan dari tubuh objek (*heat sensor*), ataupun mendeteksi seseorang dengan menggunakan *Red Laser Pointer* dan *Light Dependent Resistor*.

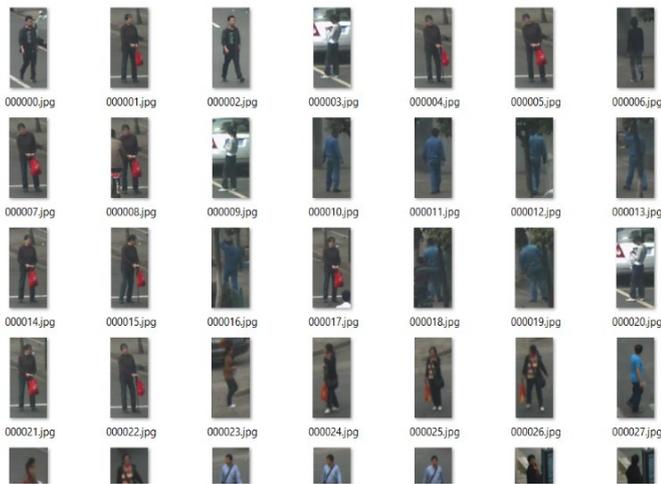
Salah satu penelitian *monitoring* ruang adalah menerapkan pengolahan citra dengan metode *adaptive motion detection* pada sistem kamera keamanan yang telah disertai *push notification* ke *smartphone* Android [2]. Hasil penelitian menunjukkan bahwa angka *false positive* deteksi gerakan di dalam ruang tergolong tinggi yakni 58,75%. Selain itu, tema *object detection* juga telah diteliti pada [2] dan [3]. Pada [2] deteksi manusia dilakukan dengan menggunakan metode *Histogram of Oriented Gradients* berbasis OpenCV. Dalam penelitian tersebut, digunakan metode HOG sebagai metode deteksi objek berupa manusia dengan mencacah *frame* pada video menjadi *frame* gambar satu-persatu. Hasil yang didapatkan pada pengujian mendeteksi objek adalah sekitar 0,00259 detik untuk objek tunggal tampak depan, 2,85291 detik untuk objek tunggal tampak samping, dan 0,00287 detik untuk mendeteksi objek tunggal tampak belakang. Selain itu objek banyak dengan posisi yang seragam membukukan waktu sekitar 2,80936 detik dan objek banyak dengan posisi bervariasi membukukan waktu deteksi sebesar 2,80042 detik.

Dari permasalahan di atas maka dilakukan sebuah penelitian untuk merancang sebuah sistem yang dapat digunakan untuk melakukan deteksi pergerakan objek berupa manusia.

II. MATERIAL DAN METODOLOGI PENELITIAN

A. Pengumpulan Data

Pada penelitian ini data yang digunakan sebagai data latih ataupun *input* layer untuk CNN adalah citra gambar manusia, didapatkan dengan pencarian *dataset* dengan menggunakan Google Search. *Dataset* didapat dari <http://mmlab.ie.cuhk.edu.hk>. Citra gambar yang digunakan sebanyak 1604 gambar. Karakteristik dari citra gambar yang digunakan adalah *portrait* di mana ukuran panjang citra lebih kecil dibandingkan dengan lebarnya. Pada *dataset* ini terdapat citra manusia yang terdiri dari berbagai sudut pengambilan gambar, berbagai jenis kelamin, berbagai umur, beragam warna pakaian, dan beragam aktivitas.

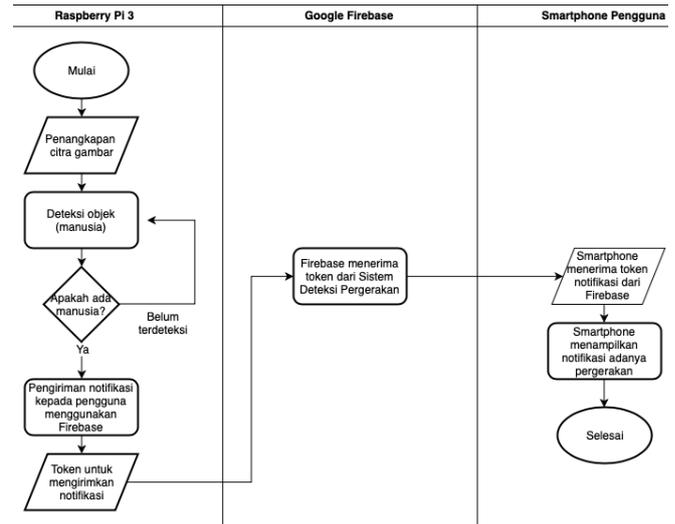


Gambar 1. Contoh *Dataset* Yang Digunakan Untuk *Training*

B. Desain Sistem

Pada tahapan ini dibuat perancangan desain sistem dengan menggunakan YOLO *Object Detection* sebagai arsitektur untuk mendeteksi adanya manusia dalam citra gambar. Sistem kerja deteksi ini adalah pada saat Raspberry Pi 3 diaktifkan, maka sistem akan menangkap video secara terus menerus. Video yang telah ditangkap akan diubah menjadi potongan *frame*. *Frame* inilah yang akan dijadikan penentu apakah ada manusia pada citra gambar masukan dari kamera. Ketika ditemukan adanya citra gambar manusia, maka sistem akan mengirimkan pemberitahuan atau notifikasi kepada pengguna. Adapun notifikasi ini akan dikirimkan menuju *smartphone* pengguna dengan memanfaatkan fitur Google *Firebase* dan koneksi internet. Dalam penelitian ini, digunakan Yolo *Object Detection* versi 2 [4, 8] serta Google *Firebase* sebagai penunjang.

Diagram alir dari sistem *monitoring* ruangan ini dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Dari Sistem *Monitoring* Ruang

C. Implementasi Desain Sistem

Tahap implementasi desain sistem ini merupakan perancangan kebutuhan pada *hardware* Raspberry Pi 3 dengan menggunakan Python dan OpenCV [4, 5, 6, 7]. Dibutuhkan modul kamera yang tersambung dengan Raspberry Pi 3 yang dihubungkan melalui satu kabel fleksibel.

Raspberry Pi 3 telah dilengkapi dengan LAN Port yang dihubungkan dengan laptop guna kebutuhan konfigurasi serta pengamatan yang dilakukan. Raspberry Pi juga telah dilengkapi dengan MicroUSB *port* sebagai input untuk sumber daya listrik, pengguna menggunakan *port* ini untuk mengisikan daya Raspberry Pi 3.

Kemudian dilakukan perancangan *user interface* yang digunakan pengguna untuk melakukan *login* pada aplikasi di *smartphone*. Aplikasi *smartphone* yang dirancang bertujuan untuk menyampaikan *push notification* kepada pengguna ketika sistem Raspberry Pi mendeteksi adanya objek manusia. *User interface* yang dirancang memiliki beberapa bagian, yaitu halaman *login*, halaman *home*, halaman *more*, halaman *profile*, serta halaman *about*.

Setelah tahap desain sistem dan desain antarmuka selesai dirancang, tahap selanjutnya adalah implementasi desain sistem. Dalam proses pembuatan sistem deteksi pergerakan objek diperlukan beberapa detik agar sistem ini dapat berjalan dengan baik serta sesuai dengan fungsinya.

D. Proses Pelatihan

Pada penelitian ini akan dilakukan pelatihan menggunakan 1604 gambar. Dua dari 1604 citra tersebut terdapat dua manusia, sehingga total keseluruhan anotasi yang dilakukan adalah 1606. Proses *training* ini berfungsi untuk melatih sistem dalam mendeteksi adanya manusia pada citra gambar. Langkah pertama yang dilakukan untuk training adalah dengan memberikan anotasi. Pemberian anotasi ini dilakukan dengan tujuan sistem mampu mengenali objek yang dimaksud dalam setiap citra yang digunakan sebagai data latih pada penelitian ini. Contoh anotasi dapat dilihat pada Gambar 3.



Gambar 3. Contoh Anotasi Pada Citra *Training*

Pada YOLO v2 proses *training* dilakukan dengan cara menerapkan jaringan syaraf tunggal pada keseluruhan citra. Jaringan ini akan membagi gambar menjadi bagian-bagian dan selanjutnya akan memprediksi kotak pembatas dan probabilitas, untuk setiap kotak bagian pembatas ditimbang probabilitasnya untuk proses klasifikasi sebagai objek atau bukan. Pada bagian akhir, akan dipilih kotak pembatas dengan nilai tertinggi untuk dijadikan sebagai pemisah objek satu dengan yang lain. YOLO v2 menggunakan arsitektur Darknet-19 yang merupakan arsitektur yang relatif sama dengan CNN lainnya yang memiliki lapisan *convolutional* serta 38 lapisan *max-pooling*. YOLO v2 menggunakan 19 lapisan *convolutional* dan 5 lapisan *max-pooling* yang diakhiri dengan lapisan *softmax*.

E. Proses Pengiriman *Push Notification*

Pada penelitian ini dirancang sebuah API yang berfungsi sebagai penghubung antara *server* Google *Firestore Cloud Messaging* (FCM) dan basis data, dengan aplikasi penerima *push notification* yang ter-*install* pada *smartphone* Android pengguna. Untuk mengirimkan notifikasi kepada pengguna, Google FCM membutuhkan *token* untuk mengirimkan notifikasi pada *smartphone* pengguna secara spesifik. Ketika pengguna melakukan *login* pada aplikasi penerima notifikasi di *smartphone*, aplikasi akan mencatatkan “*device_id*” *smartphone* pengguna ke dalam *database*, serta mencatatkan “*device_id*” tersebut pada *server* Google FCM untuk mendapatkan *token*.

Token inilah yang akan menjadi dasar FCM untuk mengirimkan notifikasi. Untuk mengurangi adanya perilaku *spamming* yang dilakukan sistem ketika adanya objek yang sama terulang dideteksi secara terus-menerus, ditambahkan *rule* pada sistem. *Rule* yang dirancang adalah notifikasi dikirimkan kepada pengguna dengan jeda 5x deteksi objek, tidak berlaku untuk deteksi pertama kali ketika sistem diaktifkan. Dengan adanya *rule* demikian maka sistem hanya akan mengirimkan notifikasi setiap adanya 5 objek yang terdeteksi dan diharapkan dengan penambahan *rule* ini mampu mengurangi perilaku *spamming*.

F. Uji Coba dan Evaluasi Sistem

Untuk uji dan evaluasi sistem dilakukan dengan menggunakan pengujian sensitivitas serta pengujian spesifisitas. Pengujian sensitivitas adalah metode perhitungan akurasi dengan membandingkan jumlah klasifikasi tepat pada suatu *class* dengan keseluruhan klasifikasi yang terdapat pada *class* tersebut. Sedangkan pengujian spesifisitas adalah metode perhitungan dengan membandingkan jumlah klasifikasi yang tidak berhubungan dengan sebuah *class* namun dianggap tepat dengan seluruh klasifikasi yang tidak berhubungan dengan *class* tersebut. Dalam perhitungan selanjutnya akan digunakan *matrix confusion* dalam perhitungan dari beberapa *class*. Untuk contoh bentuk dari *Matrix Confusion* dapat dilihat pada Tabel 1.

Tabel 1. *Matrix Confusion* Untuk Klasifikasi

Objek	Klasifikasi	
	1	2
1	TP	FN
2	FP	FN

Pada Tabel 1 terdapat kolom objek yang merupakan *class* dari data yang digunakan, serta kolom hasil klasifikasi. Apabila ada manusia dan sistem mendeteksi benar manusia, maka akan dikelompokkan menjadi *True Positive* (TP) dikarenakan hasil klasifikasi sesuai dengan yang diharapkan. Pengelompokkan ini didasarkan pada hasil deteksi menunjukkan benar manusia (*true*) dan hasil klasifikasi sesuai dengan yang seharusnya (*positive*).

Apabila objek bukan manusia teridentifikasi sebagai manusia, maka akan dikelompokkan menjadi *False Positive* (FP), dikarenakan hasil deteksi tidak sesuai dengan yang diharapkan. Pengelompokkan ini didasarkan pada hasil deteksi menunjukkan manusia, padahal objek tersebut bukanlah manusia. Ketika objek manusia dideteksi sebagai bukan manusia, maka akan dikelompokkan sebagai *False Negative*. Apabila sistem mendeteksi objek bukan manusia sebagai bukan manusia, maka akan dikelompokkan menjadi *True Negative* (TN). Pengelompokkan ini berdasarkan fakta bahwa sistem benar mendeteksi objek bukan manusia sebagai bukan manusia, maka akan dianggap *true*, serta dikarenakan yang dideteksi bukanlah manusia, maka akan dianggap *negative*. Untuk mendapatkan hasil akurasi, maka dibutuhkan rumus yang dipakai sebagai acuan dalam menghitung tingkat akurasi (1).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Untuk menghitung sensitivitas digunakan rumus (2).

$$Sensitivitas = \frac{TP}{TP + FN} \tag{2}$$

Untuk menghitung spesifisitas digunakan rumus (3).

$$Spesifisitas = \frac{TN}{TN + FP} \tag{3}$$

III. HASIL DAN PEMBAHASAN

A. Pembentukan Model Deteksi

Pembentukan model ialah langkah yang ditempuh untuk membentuk model atau *weight* dari sistem deteksi menggunakan YOLO v2 yang memiliki fungsi sebagai inti dari deteksi objek. Pada pembentukan model untuk YOLO v2, langkah pertama dalam membentuk model sebelum dilakukan pelatihan atau *training* adalah memberi anotasi pada 1604 citra gambar yang sebelumnya telah dikumpulkan. Setiap citra memiliki masing-masing satu *file annotation* berformat .xml yang memuat informasi antara lain *filename*, *width*, *height* dan *depth*. Lalu terdapat parameter lanjutan yaitu *xmin*, *ymin*, *xmax*, dan *ymax*. *Filename* adalah variabel nama dari sebuah citra gambar, lalu ada *width* yang merupakan ukuran panjang citra gambar, *height* adalah ukuran lebar citra gambar dan *depth* sebagai jumlah *layer* warna dari citra yang digunakan. Kemudian terdapat parameter lanjutan yaitu letak dari objek yang dimaksud dalam citra gambar tersebut.

Setelah pemberian anotasi pada setiap data latih yang dikumpulkan telah selesai, maka tahap selanjutnya yang akan dilakukan adalah proses pelatihan model. Pada penelitian ini menggunakan versi kecil dari bobot yolo yang disebut *tiny yolo weight*.

Proses *training* untuk membentuk *weight* yang akan digunakan dalam proses deteksi objek membutuhkan waktu selama 6 (enam) jam dengan iterasi sebanyak 30000. Pada gambar 3 dapat disimpulkan bahwa *error rate* pada iterasi ke 30000 telah mencapai nilai 0,82 yang mana telah berada diangka di bawah 1.

```

Command Prompt
step 29976 - loss 0.7332950830459595 - moving ave loss 0.8151702411487918
step 29977 - loss 0.9435983896255493 - moving ave loss 0.8280130559964676
step 29978 - loss 0.8725899457931519 - moving ave loss 0.832470744976136
step 29979 - loss 1.0416533946990967 - moving ave loss 0.8533890099484321
step 29980 - loss 0.8497948050498962 - moving ave loss 0.8530295894585785
step 29981 - loss 0.8157609105110168 - moving ave loss 0.8493027215638225
step 29982 - loss 0.8651756048202515 - moving ave loss 0.8508900098894654
step 29983 - loss 0.8177754878997803 - moving ave loss 0.8475785576904968
step 29984 - loss 1.0350161790847778 - moving ave loss 0.8663223198299249
step 29985 - loss 0.8266764879226685 - moving ave loss 0.8623577366391992
step 29986 - loss 0.9128828048706055 - moving ave loss 0.8674102434623399
step 29987 - loss 0.6153478622436523 - moving ave loss 0.8422040053404711
step 29988 - loss 0.6669870615005493 - moving ave loss 0.8246823109564789
step 29989 - loss 0.697857677936554 - moving ave loss 0.8119998476544865
step 29990 - loss 0.8475836515426636 - moving ave loss 0.8155582280433042
step 29991 - loss 0.8599408268928528 - moving ave loss 0.8199964879282592
step 29992 - loss 0.8331688046455383 - moving ave loss 0.8213137195999871
step 29993 - loss 0.7448387145996094 - moving ave loss 0.8136662190999494
step 29994 - loss 0.7287120819091797 - moving ave loss 0.8051708053808725
step 29995 - loss 0.7361403703689575 - moving ave loss 0.798267761879681
step 29996 - loss 0.8030014634132385 - moving ave loss 0.7987411320330368
step 29997 - loss 0.8250302672386169 - moving ave loss 0.8013700455535948
step 29998 - loss 0.6187969446182251 - moving ave loss 0.7831127354600579
step 29999 - loss 1.0639979839324951 - moving ave loss 0.8112012603073017
step 30000 - loss 0.9703400135040283 - moving ave loss 0.8271151336269744
Checkpoint at step 30000
Finish 300 epoch(es)
    
```

Gambar 4. Error Rate Saat Proses *Training*

B. Pengujian YOLO v2 Menggunakan Video

Pengujian ini dilakukan untuk meneliti apakah *weight* yang dilatih mampu mendeteksi manusia dengan baik dan juga diharapkan dengan adanya pengujian ini mampu didapatkan nilai akurasi dari *weight* yang telah dilatih dalam mendeteksi manusia pada kondisi *real*. Pengujian ini

dilakukan dengan menghitung jumlah manusia yang terdeteksi oleh sistem kemudian membandingkan jumlah tersebut dengan jumlah manusia yang terdapat pada *frame* kamera. Pengujian ini dilakukan menggunakan *threshold* sebesar 0.07, semakin besar nilai *threshold* yang digunakan, maka model akan semakin sulit untuk mendeteksi objek. Ketika penggunaan *threshold* diperkecil maka model akan semakin sensitif dalam melakukan deteksi objek dan akan berpengaruh pada proses deteksi keseluruhan. Hasil deteksi objek dapat dilihat pada Tabel 2.

Tabel 2. Hasil Deteksi Menggunakan Video 1

Objek Deteksi	Hasil Deteksi		Jumlah Data
	Terdeteksi	Tidak Terdeteksi	
Manusia	598	489	1087
Bukan Manusia	33	680	713

Dari hasil pengujian video pertama selama kurang lebih 75 menit, didapati hasil pengamatan yang disajikan pada Tabel 2. Dalam tabel tersebut didapati bahwa objek manusia terdeteksi sebanyak 598 *frame*. Sebanyak 489 *frame* yang terdapat manusia tetapi tidak dianggap sebagai manusia. Selain itu terdapat juga objek bukan manusia yang terdeteksi sebagai manusia sebanyak 33 *frame*. Bukan manusia dan terdeteksi bukan manusia sebanyak 680 *frame*.

Untuk memastikan pengujian menggunakan video yang telah direkam sebelumnya. Pengujian dilakukan dengan menggunakan video kedua yang berdurasi 1 menit dengan hasil seperti Tabel 3.

Tabel 3. Hasil Deteksi Menggunakan Video

Objek Deteksi	Hasil Deteksi		Jumlah Data
	Terdeteksi	Tidak Terdeteksi	
Manusia	611	443	1054
Bukan Manusia	39	707	746

Dalam tabel tersebut didapati bahwa objek manusia terdeteksi sebanyak 611 *frame*, 443 *frame* yang terdapat manusia tetapi tidak dianggap sebagai manusia. Terdapat juga objek bukan manusia tetapi terdeteksi manusia sebanyak 39 *frame*. Bukan manusia dan terdeteksi bukan manusia sebanyak 707 *frame*. Dari hasil pengujian di atas, maka nilai akurasi, sensitivitas serta spesifisitas pengujian menggunakan *weight* YOLO v2 yang sebelumnya telah dilatih disajikan pada Tabel 4.

Tabel 4. Hasil Nilai Akurasi, Sensitivitas Serta Spesifisitas

Video	TP	TN	FP	FN	Akurasi	Sensitivitas	Spesifisitas
1	598	680	33	489	71%	55,01%	95%
2	611	707	39	443	73,2%	57,96%	94,77%
Hasil Akhir					72,1%	56,49%	94,9%

C. Pengujian YOLO v2 Menggunakan Video

Pengujian yang dilakukan berikutnya adalah menguji aplikasi antarmuka Android yang digunakan sebagai aplikasi untuk memberikan *push notification* ketika sistem mendeteksi adanya objek manusia yang terdeteksi menggunakan Raspberry Pi 3 beserta dengan modul kameranya. Hasil pengujian yang dilakukan selama 5 menit, sistem mampu mengirimkan notifikasi berupa *push notification* pada *smartphone* pengguna. Dari 5 menit pengujian yang dilakukan, sistem mendeteksi adanya 84 kali manusia yang tertangkap dalam *frame*, namun pada saat pengiriman notifikasi ke *smartphone* pengguna, 6 deteksi manusia tidak ditampilkan pada notifikasi *smartphone*. Hal ini disebabkan karena *resource hardware* dari Raspberry Pi 3 telah bekerja cukup berat dikarenakan beban dalam melakukan proses pengambilan video, ditambah dengan mendeteksi objek manusia menggunakan metode YOLO v2 yang menghabiskan cukup banyak *resource* sehingga kerja prosesor selalu di atas 90% dari keseluruhan kemampuan, sehingga terjadi *delay* ketika sistem Raspberry Pi mengirimkan *request* untuk menjalankan *push notification*.

IV. KESIMPULAN

Dari penelitian yang sudah dilakukan, dapat ditarik kesimpulan dari penelitian bahwa bobot YOLO v2 yang dilatih dapat digunakan untuk mendeteksi objek berupa manusia secara *realtime* dengan nilai akurasi sebesar 72,1%, sensitivitas sebesar 56,49% dan spesifisitas sebesar 94,9%. (Tabel 4). Ke depan, pengembangan penelitian dapat dilakukan dengan meningkatkan jumlah data latih untuk

melatih bobot YOLO v2 sehingga meningkatkan akurasi dari sistem deteksi.

REFERENSI

- [1] Delia, R.P. (2009). Analisis Determinan Penyebab Timbulnya Fear of Crime Pada Kasus Pencurian Di Kalangan Ibu Rumah Tangga. *Indonesian Journal of Criminology*, 5(1).
- [2] Sikumbang, S., & Suryadi, K. (2015). Human Detection Menggunakan Metode Histogram Of Oriented Gradients (Hog) Berbasis Open CV. *Jurnal Pendidikan Teknik Elektro*, 4(2).
- [3] Umam, K. & Negara, B.S., (2016). Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, 2(2), pp.31-40.
- [4] Du, J., (2018). Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series*, Vol. 1004, No. 1, p. 012029.
- [5] Bradski, G. & Kaehler, A. (2008). *Learning OpenCV, 1st edition*, United States: O'Reilly Media.
- [6] Solem, J.E. (2012). *Programming Computer Vision with Python*, Mountain View: O'Reilly.
- [7] Rosebrock, A. (2016). *Practical Python and OpenCV*, 3rd edition, PyImageSearch.
- [8] Redmon, J., Divvala, S., Girshick, R. & Farhadi, A (2015). *You Only Look Once: Unified, Real-Time Object Detection*, Diakses <https://pjreddie.com/media/files/papers/yolo.pdf> pada tanggal 16 Februari 2019.