SISTEM PENGHITUNG JUMLAH KERUMUNAN ORANG DENGAN METODE YOLO (*YOU ONLY LOOK ONCE*)

TUGAS AKHIR

Program Studi
S1 TEKNIK KOMPUTER

Oleh:
ADIKRISNA PRADIPTYA
18410200039

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2023

SISTEM PENGHITUNG JUMLAH KERUMUNAN ORANG DENGAN METODE YOLO (*YOU ONLY LOOK ONCE*)

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik

Disusun Oleh:

Nama : **Adikrisna Pradiptya**
NIM : **18410200039**
Program Studi : **S1 Teknik Komputer**

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2023

TUGAS AKHIR

SISTEM PENGHITUNG JUMLAH KERUMUNAN ORANG DENGAN METODE YOLO (*YOU ONLY LOOK ONCE*)

Dipersiapkan dan disusun oleh:

Adikrisna Pradiptya

NIM : 18410200039

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 25 Januari 2023

Susunan Dewan Pembahas

Pembimbing:

I. **Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN: 0716117302

Digitally signed by Heri Pratikno
Date: 2023.01.30 14:13:52 +07'00'

II. **Harianto, S.Kom., M.Eng.**

NIDN: 0722087701

cn=Harianto Harianto, o=Universitas Dinamika, ou=Prodi S1 Teknik Komputer, email=hari@dinamika.ac.id, c=ID
2023.01.30 15:57:20 +07'00'

Pembahas:

Pauladie Susanto, S.Kom., M.T.

NIDN: 0729047501

Universitas Dinamika
2023.01.31 06:00:48 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Digitally signed by Universitas Dinamika
Date: 2023.01.31 13:10:37 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

Hidup adalah perjuangan tanpa henti

Terima kasih atas segala macam bentuk dukungan yang telah ayah dan ibu berikan, Tugas Akhir ini demi kalian berdua.

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama	: **Adikrisna Pradiptya**
NIM	: **18410200039**
Program Studi	: **S1 Teknik Komputer**
Fakultas	: **Fakultas Teknologi dan Informatika**
Jenis Karya	: **Laporan Tugas Akhir**
Judul Karya	: **SISTEM PENGHITUNG JUMLAH KERUMUNAN ORANG DENGAN METODE YOLO (*YOU ONLY LOOK ONCE*)**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (***Non-Exclusive Royalty Free Right***) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (***database***) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 9 Desember 2022

Adikrisna Pradiptya
NIM : 18410200039

ABSTRAK

Kemajuan teknologi yang semakin berkembang dapat membantu berbagai pekerjaan manusia menjadi lebih mudah. Teknologi *computer vision* dan *deep learning* yang dapat diimplementasikan dengan baik akan banyak membantu berbagai pekerjaan manusia menjadi lebih efektif dan efisien. Sistem pengawas banyak digunakan untuk memantau kerumunan kegiatan manusia. Mengetahui jumlah pasti orang-orang yang berada di tempat umum merupakan suatu hal yang penting untuk memastikan keselamatan publik atau kualitas layanan. Pada Tugas Akhir ini dirancang sebuah sistem penghitung jumlah kerumunan orang dengan metode YOLO (*You Only Look Once*). YOLO bekerja dengan cara melihat serta memproses suatu citra dengan hanya sekali proses. Dengan metode ini sistem mampu mendeteksi dan menghitung jumlah orang dalam kerumunan dengan cukup baik, cepat, dan akurat. Dari hasil pengujian yang telah dilakukan, sistem mampu menghitung jumlah kerumunan orang dengan baik pada kerumunan dengan tingkat rendah sampai sedang. Semakin rapat dan padat tingkat kerumunan, maka semakin besar tingkat *error* sistem. Tingkat akurasi sistem dalam menghitung jumlah kerumunan orang adalah 80.96 % dan tingkat *error*-nya 19.04 %.

Kata kunci: *Computer vision*, *deep learning*, kerumunan, sistem penghitung, YOLO.

KATA PENGANTAR

Dengan mengucapkan puji dan syukur atas kehadirat Tuhan Yang Maha Esa yang telah memberikan limpahan berkah dan rahmat-Nya, serta segala karunia yang telah diberikan kepada penulis, sehingga penulis dapat merampungkan Laporan Tugas Akhir yang berjudul "SISTEM PENGHITUNG JUMLAH KERUMUNAN ORANG DENGAN METODE YOLO (*YOU ONLY LOOK ONCE*)*"*. Laporan Tugas Akhir ini disusun dalam rangka memenuhi salah satu prasyarat dalam menyelesaikan Program Sarjana Teknik Komputer di Universitas Dinamika.

Pada kesempatan yang telah diberikan ini, penulis mengucapkan rasa terima kasih terhadap semua pihak yang memberikan dukungan dan juga saran serta bimbingan dalam upaya untuk menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Orang Tua penulis, yang telah memberikan kontribusi besar berupa dukungan penuh atas apa yang akan penulis lakukan dan juga atas apa yang telah penulis lakukan sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing I yang selalu memberikan waktu dan bimbingan serta ilmu dalam menyelesaikan Tugas Akhir beserta laporan ini.

5. Bapak Harianto, S.Kom., M.Eng., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
6. Seluruh Dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu dan juga bimbingan yang berharga dari semester 1 hingga sampai saat ini.
7. Seluruh rekan-rekan S1 Teknik Komputer yang telah memberikan dukungan dan semangatnya untuk membantu penulis untuk menyelesaikan Laporan Tugas Akhir ini.
8. Dan seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya Laporan Tugas Akhir ini.

Penulis memahami bahwa Laporan Tugas Akhir ini belum mencapai sempurna, dan masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih terdapat kesalahan baik dalam penulisan maupun bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 11 Januari 2023

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi yang semakin berkembang dapat membantu berbagai pekerjaan manusia menjadi lebih mudah. Dengan teknologi *computer vision* sebuah sistem dapat melihat dan mengenali suatu objek pada sebuah gambar maupun video layaknya penglihatan yang ada pada manusia. Dengan menggunakan *deep learning* memungkinkan proses pengenalan objek lebih mudah dan efektif. Jika teknologi ini dapat diimplementasikan dengan baik maka akan banyak membantu berbagai pekerjaan manusia menjadi lebih efektif dan efisien.

Saat ini sistem pengawas banyak digunakan untuk memantau kegiatan manusia di ruang publik maupun ruang privat seperti di bandara, pusat perbelanjaan, restoran, gedung, rumah, jalan, ataupun tempat lainnya yang membutuhkan sistem pemantauan. Mengetahui jumlah pasti orang-orang yang berada di gedung atau di tempat-tempat umum merupakan suatu hal yang penting dalam aplikasi pengawasan publik dimana jumlah orang perlu dipantau untuk memastikan keselamatan publik atau kualitas layanan. Dengan mengetahui informasi jumlah pengunjung maka pihak-pihak yang terkait dengan operasional suatu tempat dapat membuat kebijakan yang mengoptimalkan penggunaan sumberdaya untuk kebutuhan operasional tempat tersebut.

Dalam penelitian sebelumnya telah dibuat sistem yang mampu mendeteksi dan menghitung jumlah pengunjung di restoran secara otomatis saat keadaan ramai dengan tingkat akurasi penghitungan 86%. Sistem tersebut menggunakan metode *Single Shot Detector* (SSD) dengan proses pengolahan citra digital (Admaja, 2021). Selain itu dalam penelitian lain telah dibuat sistem dengan metode Faster R-CNN yang mampu mendeteksi kapasitas orang dalam ruangan. Sistem tersebut mampu mendeteksi orang yang berada di POV (*Point of View*) kamera dengan tingkat akurasi 77% (Laili, 2022). Dari beberapa penelitian tersebut ada keterbatasan yaitu orang dapat dideteksi jika orang yang dideteksi tidak berhimpitan dan tidak terhalang orang lain.

Berdasarkan latar belakang tersebut maka dalam Tugas Akhir ini dirancang sebuah sistem penghitung jumlah kerumunan orang dengan metode YOLO (*You Only Look Once*) yang mampu menghitung jumlah orang dalam kerumunan dengan cepat dan akurat. Hasil penelitian ini dapat digunakan untuk membantu pihak yang terkait dalam sistem pengawas untuk keperluan keselamatan publik maupun peningkatan kualitas layanan.

1.2 Rumusan Masalah

Rumusan masalah dari Tugas Akhir ini berdasarkan latar belakang diatas adalah bagaimana tingkat akurasi penghitungan jumlah kerumunan orang dengan metode YOLO (*You Only Look Once*)?

1.3 Batasan Masalah

Pembahasan masalah dalam Tugas Akhir ini dibatasi pada hal-hal sebagai berikut:

1. Data tidak diambil dari kamera pengawas yang disiarkan secara langsung (*realtime*) tetapi menggunakan hasil foto.
2. Pencahayaan cukup dan tidak gelap.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah merancang sistem penghitung jumlah kerumunan orang dengan metode YOLO (*You Only Look Once*) secara akurat.

1.5 Manfaat

Manfaat yang dapat diperoleh dari Tugas Akhir ini adalah sebagai berikut:

1. Memahami sistem penghitung jumlah kerumunan orang dengan metode YOLO (*You Only Look Once*).
2. Membantu pekerjaan manusia dalam hal yang terkait dengan sistem pengawas untuk keselamatan publik dan kualitas layanan.

BAB II
LANDASAN TEORI

2.1 Kerumunan

Kerumunan atau *crowd* adalah adalah sekumpulan orang yang bersama-sama ada di suatu tempat yang biasanya bersifat sementara, tidak teratur, dan terjadi secara spontan. Menurut (Kurniasih, 2021) kerumunan dapat dibagi menjadi beberapa jenis, yaitu:

a. Kerumunan formal, yaitu kerumunan yang memiliki pusat perhatian yang sama. Contoh: penonton sepak bola dan penonton bioskop.
b. Kerumunan terencana yang ekspresif, yaitu kerumunan yang tidak memiliki pusat perhatian yang sama namun memiliki tujuan yang sama. Contoh: orang dalam tempat wisata dan orang dalam pesta.
c. Kerumunan santai tetapi tidak nyaman, artinya kerumunan yang terbentuk karena kebutuhan akan ruang publik. Contoh: orang menunggu antrian dan orang menunggu kereta.
d. Kerumunan panik, artinya kerumunan yang terbentuk karena kepanikan akan suatu bahaya. Contoh: kerumunan di titik evakuasi bencana alam.
e. Kerumunan melawan hukum, artinya kerumunan yang terbentuk karena adanya tindakan melawan hukum. Contoh: tawuran dan pengeroyokan.

2.2 Sistem Pengawas

Sistem pengawas atau *Surveillance System* atau disebut CCTV (*Closed Circuit Television System*) digunakan untuk memantau semua kegiatan secara visual (*audio visual*) dengan memasang suatu alat berupa kamera pada area tertentu. Sistem ini berfungsi untuk mengawasi dan merekam suatu kejadian di suatu area tertentu. CCTV terdiri dari kamera, *video recorder*, dan monitor yang terintegrasi dalam suatu sistem jaringan *online*. Penggunaan CCTV bertujuan untuk memantau area yang luas ataupun lokasi yang sulit diakses dalam waktu bersamaan (Grahafatta.com, 2022).

2.3 *Computer Vision*

Computer vision merupakan jenis kecerdasan buatan pada komputer untuk belajar berdasarkan hasil penglihatannya untuk kemudian mendapatkan pemahaman dari hasil penglihatan tersebut. *Computer vision* memanfaatkan citra digital atau video. Pembelajaran pada komputer secara mendalam (*deep learning*) biasa menggunakan citra digital atau video. Objek akan dideteksi dan diidentifikasi secara akurat kemudian akan diproses atau dilakukan suatu aksi. Kamera CCTV biasa digunakan untuk menangkap citra digital atau video (Rusdi, 2022).

Cara kerja *computer vision* adalah dengan menangkap gambar-gambar yang dijadikan model. Model disimpan ke dalam memori. Data dipelajari oleh sistem komputer. Komputer membandingkan gambar yang ditangkap berikutnya berdasarkan model sebelumnya, lalu mengambil kesimpulan dari gambar tersebut. Salah satu kunci dalam *computer vison* adalah semakin banyak gambar yang dapat dijadikan sebagai sumber akan mempertinggi tingkat keakuratan prediksi dari komputer itu sendiri.

2.4 *Deep Learning*

Deep learning merupakan bagian dari kecerdasan buatan (*Artificial Intelligence*) dan juga *machine learning*. *Machine learning* adalah sekumpulan algoritma yang telah dilatih berdasarkan data yang ada. *Artificial Intelligence* (AI) adalah kemampuan komputer yang memungkinkan untuk meniru perilaku manusia. Struktur otak manusia merupakan inspirasi dari algoritma *deep learning*. *Deep learning* akan berusaha membuat kesimpulan berdasarkan struktur logika tertentu sebagaimana halnya pada manusia dalam menganalisis data.

Struktur algoritma *multilayer* atau *neural networks* digunakan dalam algoritma *deep learning*. Rancangan *neural network* ini berdasarkan pada struktur otak manusia. *Neural network* dapat dilatih untuk melakukan tugas seperti mengidentifikasi pola dan mengklasifikasikan berbagai jenis informasi sama seperti manusia dalam menggunakan otaknya. Kita dapat mengerjakan berbagai tugas seperti klasifikasi, regresi, atau pengklasteran (*clustering*) dengan menggunakan *neural network* (Asy'ari, 2020).

2.5 YOLO

YOLO (*You Only Look Once*) merupakan sebuah algoritma untuk melakukan deteksi secara *realtime*. Berbeda dengan kebanyakan algoritma sistem deteksi lainnya, YOLO menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi *bounding box* (kotak pembatas) dan probabilitas. Setiap kotak pembatas dihitung probabilitasnya untuk pengklasifikasian sebagai objek atau bukan. Pada bagian akhir akan dipilih kotak pembatas dengan nilai yang paling tinggi untuk dijadikan sebagai pemisah objek satu dengan yang lain (Dzulfikar, 2018).

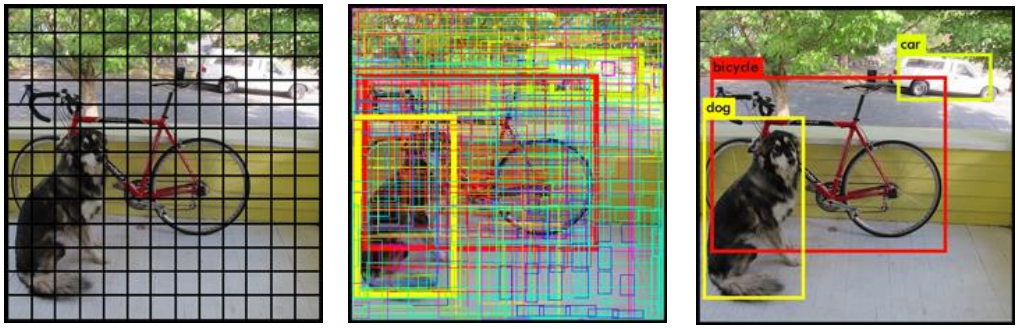
Gambar 2.1 YOLO Real Time Object Detection
(Sumber: https://github.com/pjreddie/darknet/)

Algoritma YOLO merupakan algoritma untuk pendeteksian objek. Sesuai dengan sebutannya *You Only Look Once*, cara kerja algoritma tersebut yaitu melihat serta memproses suatu citra (*image*) dengan hanya sekali proses. YOLO adalah algoritma berbasis *regression* (regresi) yang dalam dalam satu proses menghasilkan keluaran berupa prediksi kelas dan *bounding box* (kotak pembatas) dalam setiap objek (Alfarisi, 2020).

Layer	kernel	stride	output shape
Input			(416, 416, 3)
Convolution	3x3	1	(416, 416, 16)
MaxPooling	2x2	2	(208, 208, 16)
Convolution	3x3	1	(208, 208, 32)
MaxPooling	2x2	2	(104, 104, 32)
Convolution	3x3	1	(104, 104, 64)
MaxPooling	2x2	2	(52, 52, 64)
Convolution	3x3	1	(52, 52, 128)
MaxPooling	2x2	2	(26, 26, 128)
Convolution	3x3	1	(26, 26, 256)
MaxPooling	2x2	2	(13, 13, 256)
Convolution	3x3	1	(13, 13, 512)
MaxPooling	2x2	1	(13, 13, 512)
Convolution	3x3	1	(13, 13, 1024)
Convolution	3x3	1	(13, 13, 1024)
Convolution	1x1	1	(13, 13, 125)

Gambar 2.2 Arsitektur YOLO
(Sumber: https://machinelearning.mipa.ugm.ac.id/yolo-you-only-look-once/)

Dalam algoritma YOLO, proses pertama adalah melakukan input sebuah *image* atau video. *Image* tersebut kemudian diubah ukurannya menjadi 416x416 *pixels*. Algoritma YOLO memiliki dua proses utama, c*onvolution* dan *maxpooling*. *Convolution* merupakan proses manipulasi citra menggunakan *mask* atau *subwindows* eksternal untuk menghasilkan citra baru. *Kernel* 3x3 digunakan pada proses konvolusi ini. *Maxpooling* merupakan proses pengurangan masukan secara spasial (pengurangan jumlah parameter) dengan melakukan operasi *downsampling* dengan mengambil nilai maksimum dari bagian tersebut. Dalam *maxpooling* ini digunakan *kernel* 2x2 *filters* dan *stride* 2, yang artinya setiap matriks akan selalu dibagi menjadi setengahnya (416x416 menjadi 208x208, dst.). Selanjutnya proses *convolution* dan *maxpooling* terus diulang hingga menghasilkan keluaran *grid cell* 13x13x125. 125 *channels* merupakan akhir dari proses YOLO. 125 ini berisi data untuk prediksi kelas dan kotak pembatas *(bounding box)*. Dalam setiap sel grid diprediksi 5 *bounding box* dan sebuah *bounding box* dideskripsikan dengan 25 elemen data maka terdapat 125 *channels*.



Gambar 2.3 Grid sel *image* YOLO, prediksi *bounding box*, hasil prediksi *bounding box*
(Sumber: https://medium.com)

Pada prosesnya YOLO akan membagi citra menjadi 13x13 grid sel. Dalam setiap *bounding box*, sel juga memprediksi kelas yang berfungsi sebagai klasifikasi. YOLO memberikan probabilitas pada semua kelas yang mungkin. Skor kepercayaan untuk *bounding box* dan prediksi kelas digabungkan menjadi satu hasil akhir yang memberi tahu kemungkinan bahwa *bounding box* ini berisi jenis objek tertentu. Jika terdapat 13x13 = 169 sel grid dan setiap sel memprediksi 5 *bounding box*, maka total ada 845 *bounding box*. Semua prediksi dibuat dalam sekali proses dan pada waktu yang sama meski ada 845 prediksi yang terpisah. Terdapat beberapa kotak yang memiliki skor kepercayaan rendah. Kotak dengan skor prediksi diatas 30% (ambang batas dapat diatur tergantung keakuratan yang dinginkan) yang hanya akan disimpan dan ditampilkan YOLO. Dari gambar diatas, prediksi terakhir adalah dari 845 *bounding box* hanya akan disimpan 3 karena ketiga *bounding box* memberikan hasil terbaik. Penghilangan *bounding box* yang memiliki skor kepercayaan rendah merupakan proses *non-max suppression*.

2.6 Dataset

Komputer belajar dari informasi yang diterimanya. Prinsip ini meniru pembelajaran manusia. Seseorang belajar dari kumpulan data yang darinya pengetahuan terbentuk. Akhirnya, informasi dikumpulkan sebagai informasi. Dengan inilah komputer diajari untuk menjadi cerdas. Dengan demikian, data memainkan peran penting dalam membentuk kecerdasan komputer. Dalam prosesnya, data ini berlaku sebagai input, yang disebut sebagai dataset. Dataset ini

dibagi menjadi 2 bagian yaitu sebagian menjadi *training* dataset, sebagian lagi *testing* dataset. *Training* dataset digunakan dalam pembelajaran komputer. Sedangkan *testing* dataset digunakan untuk pengujian model. Ada juga yang membagi dataset menjadi 3 bagian. Ada tambahan *validation* dataset yang digunakan untuk validasi atau evaluasi model yang dipilih. Jika membagi dataset menjadi 2 bagian dengan menghilangkan *validation* dataset, maka berarti *validation* dataset digabungkan dengan *testing* dataset (Kitainformatika.com, 2020).

Salah satu dataset yang banyak digunakan adalah COCO. COCO (*Common Objects in Context*) adalah dataset untuk *object detection, segmentation* dan *captioning* dengan skala yang sangat besar. COCO menyediakan banyak sekali fitur diantaranya adalah *object segmentation*, *recognition in context*, 330.000 *images* (>200.000 berlabel), 1.5 juta contoh objek, 80 kategori objek, dan sebagainya.

2.7 OpenCV

OpenCV (*OpenSource Computer Vision Library*) adalah *library opensource* yang banyak digunakan untuk menyederhanakan pemrograman yang berhubungan dengan citra digital. OpenCV memiliki banyak fitur seperti pengenalan wajah, deteksi wajah, pelacakan wajah, Kalman *filtering*, dan berbagai jenis metode AI (*Artificial Intellegence*). OpenCV juga menyediakan beberapa algoritma terkait *Computer Vision* sederhana. OpenCV telah mendukung bahasa pemrograman C/C++, Phyton, Java, dan Matlab (Sidharta, 2017).

2.8 Python

Python merupakan salah satu bahasa pemrograman *high level dan general purpose*. Python merupakan *high level programming* karena bahasa ini mudah digunakan atau dipahami, tersedia banyak *library*, mudah diakses, dan dapat berjalan disemua platform. Python merupakan *general purpose programming* karena dapat digunakan untuk berbagai hal seperti *machine learning, artificial intelligence, embedded applications, web development*, dan *game development*. *Python* dapat diunduh secara gratis dan bebas biaya karena bersifat *opensource*

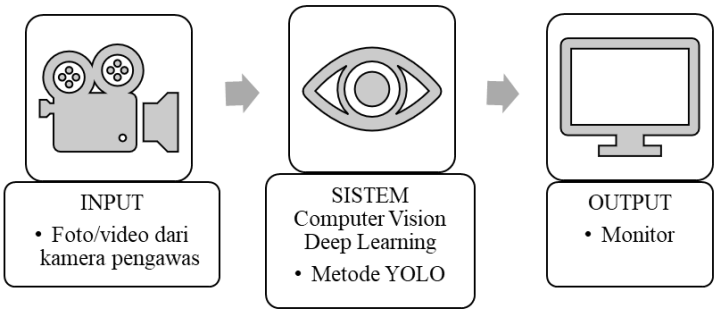
sehingga memudahkan pemula untuk mempelajarinya. *Python* merupakan bahasa pemrograman yang mudah dipelajari dan dipahami sehingga dianjurkan bagi pemula yang akan belajar pemrograman (Nurlisa, 2022).

2.9 Keras *Library*

Keras merupakan *library* berbasis *python* yang mudah digunakan, Keras biasa digunakan untuk pengembangan dan evaluasi model *deep learning*. Keras digunakan untuk menyederhanakan implementasi algoritma-algoritma *deep learning* diatas TensorFlow. Keras dikelola dan dikembangkan oleh Francois Chollet dan dirancang agar cepat dan mudah diimplementasikan. Keras menawarkan API yang konsisten dan sederhana untuk pengembangan algoritma *deep learning* (Algorit.ma, 2022).

BAB III
METODOLOGI PENELITIAN

3.1 Perancangan Perangkat Keras



Gambar 3.1 Model Perancangan *Hardware*

Gambar 3.1 diatas menunjukkan model perancangan perangkat keras sistem penghitung jumlah kerumunan orang. Input sistem berasal dari foto atau video yang diperoleh dari kamera pengawas. Kemudian data dari input diproses oleh sistem yang memanfaatkan *computer vision* dan *deep learning* dengan metode YOLO (*You Only Look Once*) untuk mendeteksi objek orang dalam kerumunan lalu kemudian dihitung jumlahnya. Output yang ditampilkan pada monitor berupa foto atau video yang menunjukkan orang yang terdeteksi dan jumlahnya.

3.2 Dataset

Dataset yang digunakan dalam sistem penghitung jumlah kerumunan orang dalam penelitian ini yaitu dataset COCO yang dapat diakses melalui *website* https://cocodataset.org. Dataset COCO terdiri dari 80 kelas kategori objek. Untuk mendeteksi orang saja maka dalam sistem hanya akan ditampilkan prediksi dari kategori kelas *person* atau orang. Untuk model YOLO dapat menggunakan Darknet YOLO yang dapat diakses melalui *website* https://pjreddie.com/darknet/yolo/. Disini didapatkan YOLO-COCO model yang terdiri dari *file* konfigurasi *.cfg* dan file bobot pelatihan *.weights* yang digunakan sebagai *pre-trained* model untuk proses *transfer learning*. Model kemudian dikonversikan menjadi model Keras .h5.

3.3 Perancangan Perangkat Lunak

Gambar 3.2 Diagram Alir Sistem Penghitung Jumlah Kerumunan Orang

Penjelasan diagram alir sistem penghitung jumlah kerumunan orang seperti pada Gambar 3.2 adalah sebagai berikut:

1. Start, Input dataset

   Program dimulai dengan input dataset dalam model YOLO sebagai *pre-trained model*. Pada penelitian ini digunakan YOLO-COCO dataset yang telah dikonversikan menjadi model Keras .h5. Lokasi dataset atau model yang digunakan dapat diatur dalam skrip kode program berikut:

```
class YOLO(object):
      _defaults = {{
           "model_path": 'model_data/yolov3.h5',
           "anchors_path": 'model_data/yolo_anchors.txt',
           "classes_path": 'model_data/coco_classes.txt',
           "score" : 0.3,
           "iou" : 0.45,
           "model_image_size" : (416, 416),
           "gpu_num" : 1,
      }
```

Dalam skrip tersebut juga dapat diatur kategori kelas, skor ketelitian, dan ukuran model yang akan digunakan. Kemudian *load* model dengan perintah:

```
self.yolo_model = load_model(model_path, compile=False)
```

2. Input *image* atau foto yang diuji

   Memasukkan data yang akan diuji berupa *image* atau foto seperti yang ditunjukkan pada Gambar 3.3 dibawah.

Gambar 3.3 Input *image* atau foto (Foto 18 input)
(Sumber: https://www.v7labs.com/open-datasets/scut-head)

Pada program ditunjukkan dengan perintah:

```
img = input('Input image filename:')
        try:
            image = Image.open(img)
```

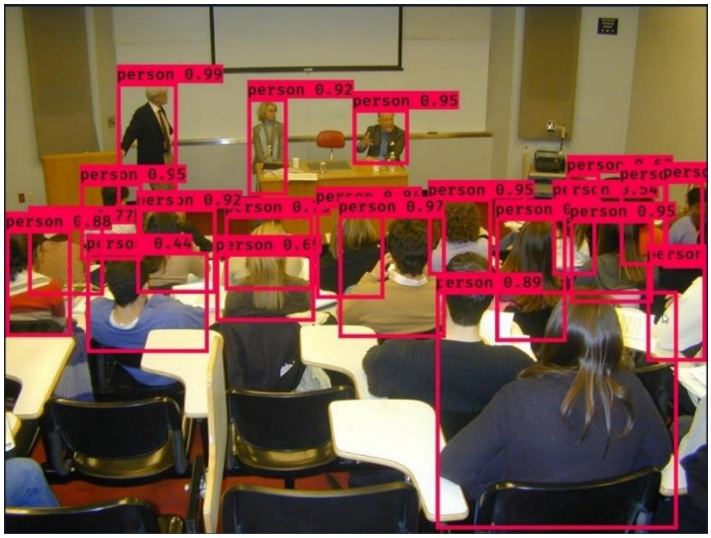
3. Deteksi orang dengan metode YOLO

   Tahap dimana sistem mencoba mengenali objek pada foto dengan metode YOLO. Data masukan yang ada akan diproses seperti proses *resize*, pembagian grid, dan penentuan letak *bounding box*. Pada program ditunjukkan dengan skrip kode berikut:

```
def detect_image(self, image):
 start = timer()
      if self.model_image_size != (None, None):
           assert self.model_image_size[0]%32 == 0,'Multiples
                                                 of 32 required'
           assert self.model_image_size[1]%32 == 0, 'Multiples
                                                 of 32 required'
           boxed_image = letterbox_image(image,
           tuple(reversed(self.model_image_size)))
      else:
           new_image_size = (image.width - (image.width % 32),
                             image.height - (image.height % 32))
        boxed_image = letterbox_image(image,new_image_size)

      image_data = np.array(boxed_image, dtype='float32')
      print(image_data.shape)
      image_data /= 255.
      image_data = np.expand_dims(image_data, 0)

      out_boxes, out_scores, out_classes = self.sess.run(
           [self.boxes, self.scores, self.classes],
           feed_dict={
               self.yolo_model.input: image_data,
               self.input_image_shape: [image.size[1],
                                        image.size[0]],
               K.learning_phase(): 0

           })
```

4. Menampilkan *bounding box* orang yang terdeteksi

   Objek orang yang terdeteksi akan diberikan *bounding box* dan skor kepercayaan seperti pada Gambar 4.4 berikut:



Gambar 3.4 Menampilkan *bounding box* dan skor kepercayaan

Pada program ditunjukkan dengan skrip kode berikut:

```
draw = ImageDraw.Draw(image)
for i, c in reversed(list(enumerate(out_classes))):
     predicted_class = self.class_names[c]
     if predicted_class == 'person':
          box = out_boxes[i]
          score = out_scores[i]

          label = '{} {:.2f}'.format(predicted_class, score)
          label_size = draw.textsize(label, font)

          top, left, bottom, right = box
          top = max(0, np.floor(top + 0.5).astype('int32'))
          left = max(0, np.floor(left + 0.5).astype('int32'))
          bottom = min(image.size[1], np.floor(bottom +
                  0.5).astype('int32'))
          right = min(image.size[0], np.floor(right +
                  0.5).astype('int32'))
          print(label, (left, top), (right, bottom))

          if top - label_size[1] >= 0:
              text_origin = np.array([left, top - label_size[1]])
          else:
               text_origin = np.array([left, top + 1])

          for i in range(thickness):
               draw.rectangle(
               [left + i, top + i, right - i, bottom - i],
               outline=self.colors[c])
          draw.rectangle(
               [tuple(text_origin), tuple(text_origin +
                                          label_size)],
               fill=self.colors[c])
          draw.text(text_origin, label, fill=(0, 255, 0),
                                               font=font)
```

5. Menghitung jumlah orang yang terdeteksi

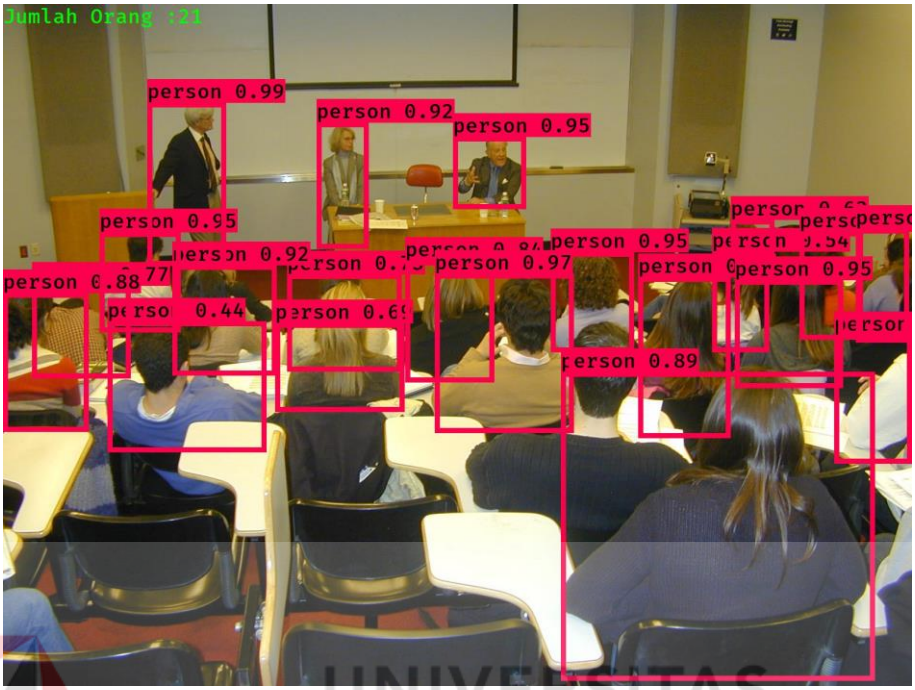
   Orang yang terdeteksi dihitung dengan menjumlahkan *bounding box* yang telah diperoleh dari proses sebelumnya. Untuk itu ditambahkan *counter* pada program menjadi seperti berikut:

```
draw = ImageDraw.Draw(image)
num_person = 0
for i, c in reversed(list(enumerate(out_classes))):
     predicted_class = self.class_names[c]
     if predicted_class == 'person':
          box = out_boxes[i]
          score = out_scores[i]

          label = '{} {:.2f}'.format(predicted_class, score)
          label_size = draw.textsize(label, font)

          top, left, bottom, right = box
          top = max(0, np.floor(top + 0.5).astype('int32'))
          left = max(0, np.floor(left + 0.5).astype('int32'))
          bottom = min(image.size[1], np.floor(bottom +
                   0.5).astype('int32'))
          right = min(image.size[0], np.floor(right +
                   0.5).astype('int32'))
          print(label, (left, top), (right, bottom))

          if top - label_size[1] >= 0:
             text_origin = np.array([left, top - label_size[1]])
          else:
               text_origin = np.array([left, top + 1])

          num_person +=1
          for i in range(thickness):
               draw.rectangle(
               [left + i, top + i, right - i, bottom - i],
               outline=self.colors[c])
          draw.rectangle(
               [tuple(text_origin), tuple(text_origin +
                                          label_size)],
               fill=self.colors[c])
          draw.text(text_origin, label, fill=(0, 255, 0),
                                                font=font)
draw.text((0,0), 'Jumlah Orang :{}'.format(num_person), fill=(0,
                                          255, 0), font=font)
```

6. Output

   Output yang dihasilkan berupa *image* atau foto yang menunjukkan orang yang terdeteksi yang ditandai dengan *bounding box* dan label berupa kelas *person* dan skor kepercayaan.

Selain itu juga ditampilkan jumlah kerumunan orang yang terdeteksi. Output sistem ditunjukkan seperti pada Gambar 3.5 dibawah.



Gambar 3.5 Tampilan output sistem (foto 18 output)

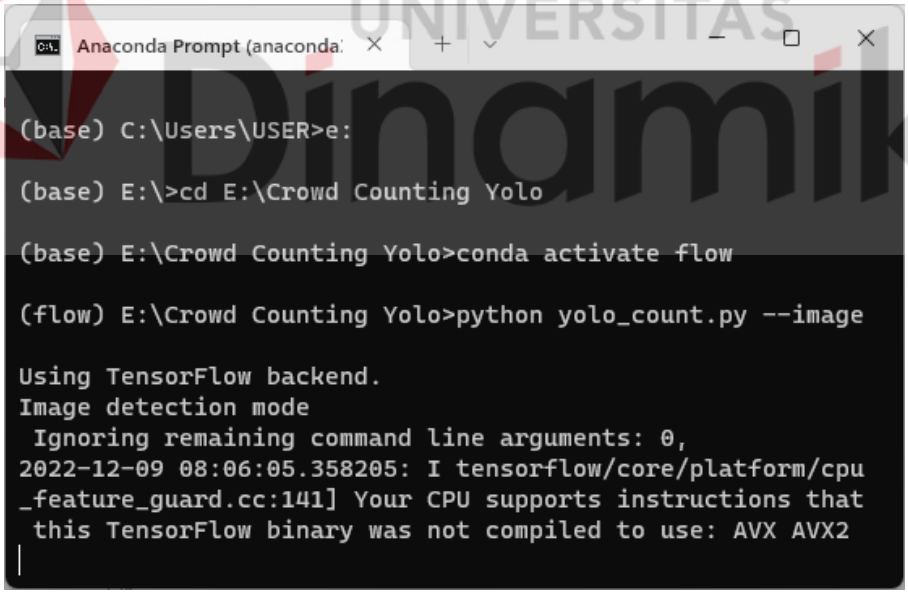
BAB IV
HASIL DAN PEMBAHASAN

4.1 Tujuan Pengujian

Tujuan dari pengujian ini adalah untuk mengetahui tingkat akurasi dari sistem penghitung jumlah kerumunan orang. Penghitungan secara sistem akan dibandingkan dengan penghitungan secara manual.

4.2 Prosedur Pengujian

1. Menjalankan sistem dengan cara mengeksekusi program yang telah dibuat dalam bahasa pemrograman Python dengan Anaconda Prompt seperti pada Gambar 4.1 dibawah.



Gambar 4.1 Menjalankan sistem

2. Melakukan pengambilan data input berupa *image* atau foto.
   Berikut merupakan beberapa data input foto:

Gambar 4.2 Foto 10 input

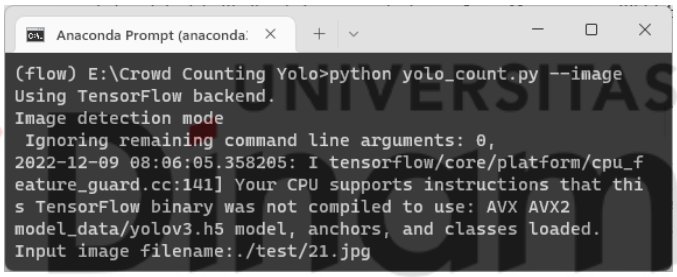
Gambar 4.3 Foto 12 input

Gambar 4.4 Foto 13 input

Gambar 4.5 Foto 17 input

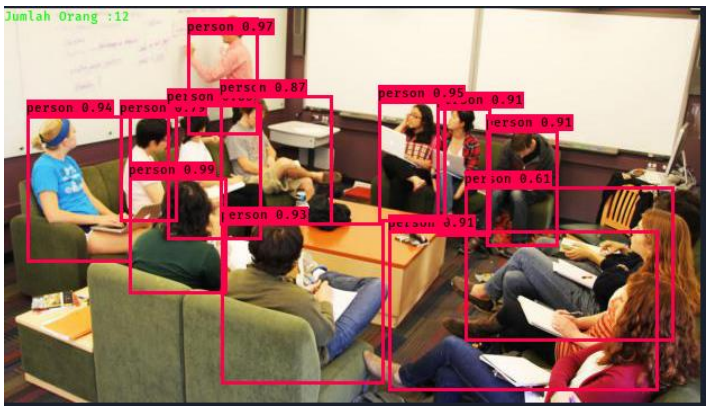
Gambar 4.6 Foto 27 input

Data pengujian lainnya berupa foto input terlampir pada Lampiran 2.

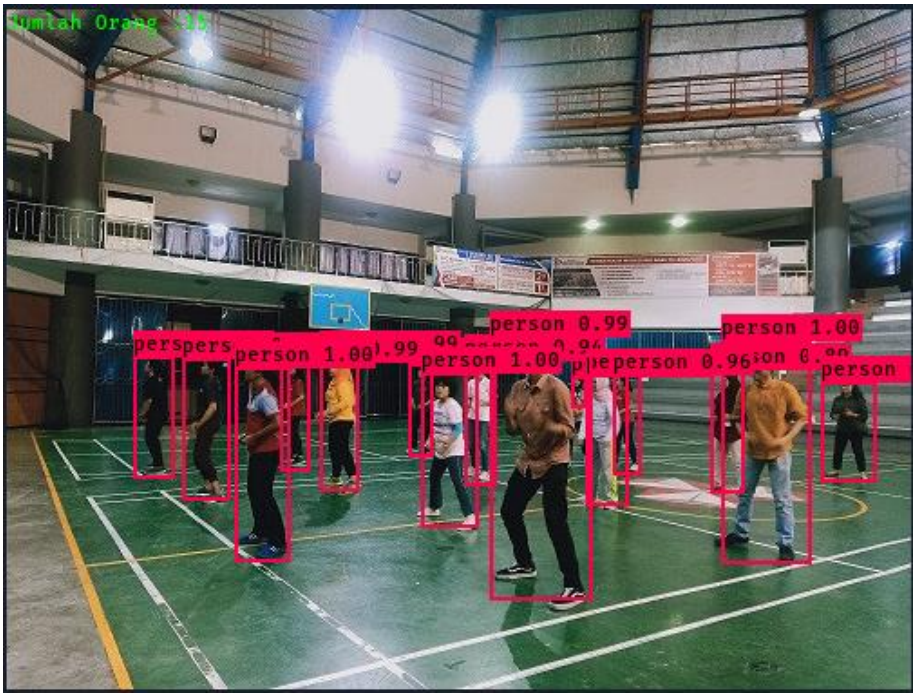


Gambar 4.7 Input *image* atau foto pada program

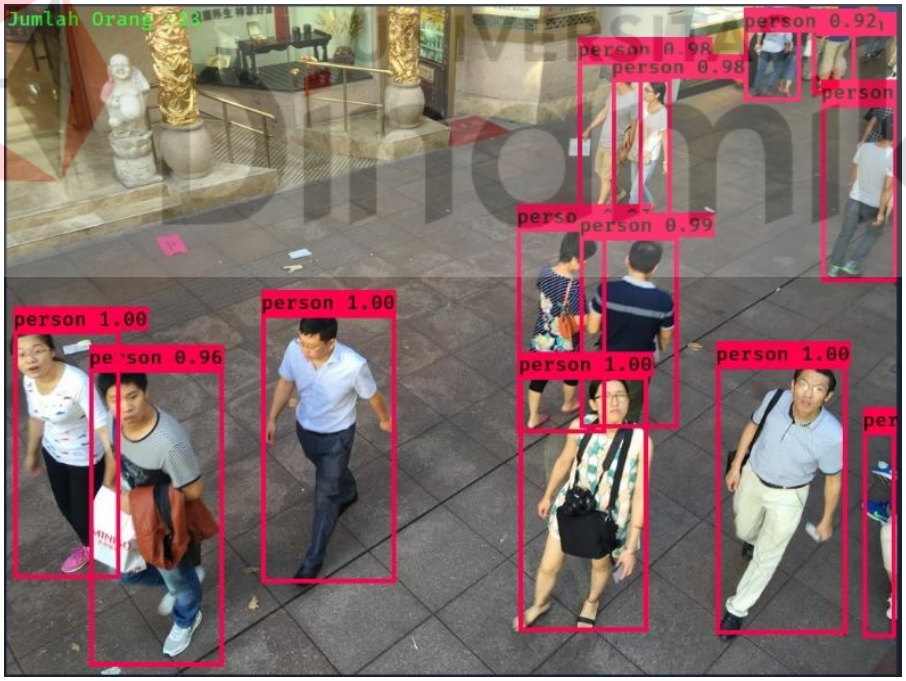
3. Menampilkan *image* atau foto orang yang terdeteksi berupa *bounding box*, skor kepercayaan, dan jumlah orang. Berikut merupakan beberapa data output foto:



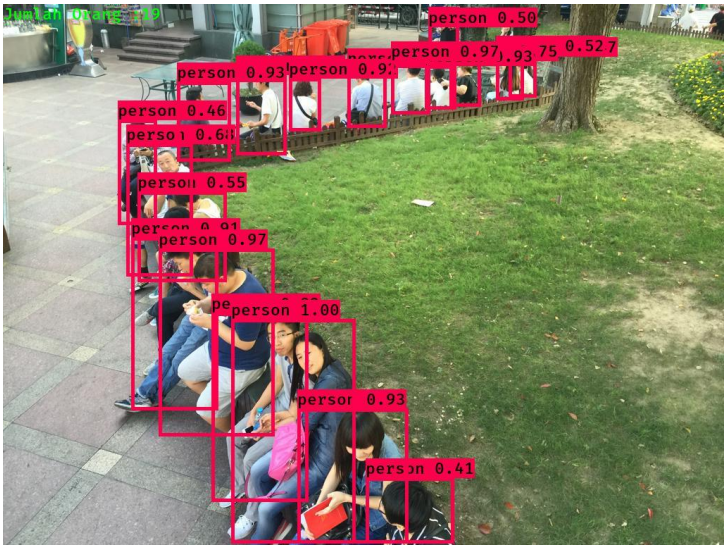
Gambar 4.8 Foto 10 output



Gambar 4.9 Foto 12 output



Gambar 4.10 Foto 13 output



Gambar 4.11 Foto 17 output

Gambar 4.12 Foto 27 output

Data pengujian lainnya berupa foto output terlampir pada Lampiran 2.

4. Melakukan penghitungan secara manual untuk mendapatkan jumlah orang yang sebenarnya.
5. Menghitung nilai *error* dan tingkat akurasi sistem dengan rumus:

$$Error\ (\%) = \frac{\text{selisih penghitungan}}{\text{penghitungan manual}}\ x\ 100\% \qquad (4.1)$$

$$\text{Akurasi}\ (\%) = \frac{\text{jumlah penghitungan sistem}}{\text{jumlah penghitungan manual}}\ x\ 100\% \qquad (4.2)$$

4.3 Hasil Pengujian

Tabel 4.1 Tabel Hasil Pengujian Penghitungan

Uji ke-	Data Uji	Penghitungan Sistem	Penghitungan Manual	Selisih Penghitungan	Error (%)
1	Foto 1	2	2	0	0
2	Foto 2	5	5	0	0
3	Foto 3	6	6	0	0
4	Foto 4	7	7	0	0
5	Foto 5	8	9	1	11.11
6	Foto 6	11	11	0	0
7	Foto 7	12	12	0	0
8	Foto 8	13	13	0	0
9	Foto 9	12	13	1	7.69
10	Foto 10	12	13	1	7.69
11	Foto 11	13	14	1	7.14
12	Foto 12	15	15	0	0
13	Foto 13	13	16	3	18.75
14	Foto 14	15	17	2	11.76
15	Foto 15	19	20	1	5
16	Foto 16	19	20	1	5
17	Foto 17	19	21	2	9.52
18	Foto 18	21	23	2	8.69
19	Foto 19	21	24	3	12.5
20	Foto 20	24	24	0	0
21	Foto 21	27	28	1	3.57
22	Foto 22	27	30	3	10
23	Foto 23	29	30	1	3.33
24	Foto 24	25	30	5	16.67
25	Foto 25	25	33	8	24.24
26	Foto 26	34	39	5	12.82
27	Foto 27	35	42	7	16.67
28	Foto 28	38	52	14	26.92
29	Foto 29	40	64	24	37.5
30	Foto 30	44	97	53	54.64
Jumlah		591	730	139	

Berdasarkan rumus (4.1) dan (4.2) didapatkan hasil perhitungan seperti pada tabel pengujian, *error* sistem, dan akurasi sistem sebagai berikut:

$$\text{Error (\%)} = \frac{\text{jumlah selisih penghitungan}}{\text{jumlah penghitungan manual}} \; x\; 100\%$$

$$= \frac{139}{730} \; x\; 100\%$$

$$= 19.04\ \%$$

$$\text{Akurasi (\%)} = \frac{\text{jumlah penghitungan sistem}}{\text{jumlah penghitungan manual}} \; x\; 100\%$$

$$= \frac{591}{730} \; x\; 100\%$$

$$= 80.96\ \%$$

BAB V
PENUTUP

5.1 Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem mampu menghitung jumlah kerumunan orang dengan baik pada kerumunan dengan tingkat rendah sampai sedang (sampai dengan sekitar 50 orang pada hasil pengujian).
2. Orang yang tidak terdeteksi dikarenakan orang dalam foto terlalu kecil, tidak jelas, buram, atau kabur serta orang yang dalam kerumunan tersebut terlalu rapat atau padat.
3. Semakin rapat dan padat tingkat kerumunan, maka semakin besar tingkat *error* sistem. Pada Tugas Akhir ini tingkat *error* dari sistem adalah 19.04 %.
4. Tingkat akurasi sistem dalam menghitung jumlah kerumunan orang adalah 80.96 %.

5.2 Saran

Pada penelitian selanjutnya penulis memiliki beberapa saran yang diharapkan dapat dikembangkan dan diimplementasikan, yaitu:

1. Mengembangkan sistem penghitungan jumlah kerumunan orang dengan metode lain misalnya MCNN (Multi-column CNN) atau CSRNet agar dapat menghitung lebih akurat untuk kerumunan yang rapat dan padat.
2. Menggunakan dataset lain yang lebih cocok dan lebih baik untuk menghitung jumlah kerumunan orang agar sistem lebih efisien dan lebih akurat.
3. Menampilkan program sistem penghitung jumlah kerumunan orang secara GUI (*Graphical User Interface)* agar mudah digunakan.

DAFTAR PUSTAKA

Admaja, Y. P. (2021). *Sistem Penghitung Jumlah Pengunjung di Restoran menggunakan Kamera Berbasis Single Shot Detector (SSD).* Surabaya: Universitas Dinamika.

Alfarisi, H. M. (2020, Maret 25). *You Only Look Once (YOLO) Algoritma Deep Learning Object Detection Terbaik*. Retrieved from medium.com: https://haiqalmuhamadalfarisi.medium.com/you-only-look-once-yolo-algoritma-deep-learning-object-detection-terbaik-af9ed81de9e9

Algorit.ma. (2022, April 12). *Kelebihan Library Keras Dalam Deep Learning*. Retrieved from Algoritma: https://algorit.ma/blog/library-keras-2022/

Asy'ari, M. Z. (2020). *Apa Itu Deep Learning dan Bagaimana Cara Kerja nya?* Retrieved from Auftechnique: https://auftechnique.com/apa-itu-deep-learning-bagaimana-cara-kerja-nya/

Dzulfikar, M. (2018, Agustus 15). *YOLO (you only look once*. Retrieved from Universitas Gadjah Mada Menara Ilmu Machine Learning: https://machinelearning.mipa.ugm.ac.id/2018/08/05/yolo-you-only-look-once/

Grahafatta.com. (2022). *Surveillance System*. Retrieved from Grahafatta.com: http://grahafatta.com/solusi-it/sistem-pengawasan-dan-keamanan/surveillance-system/

Kitainformatika.com. (2020, Juli 7). *Machine Learning dan Istilah dalam Data Set*. Retrieved from Kitainformatika.com: http://www.kitainformatika.com/2020/07/machine-learning-dan-istilah-dalam-data.html

Kurniasih, W. (2021). *Kelompok Sosial: Pengertian, Macam, Klasifikasi, Syarat, Ciri-ciri*. Retrieved from Gramedia.com: https://www.gramedia.com/literasi/kelompok-sosial/

Laili, S. N. (2022). *Sistem Deteksi Kapasitas Manusia di dalam Ruangan menggunakan Metode Faster R-CNN*. Surabaya: Universitas Dinamika.

Nurlisa, A. (2022, Mei 20). *Pemula Wajib Tahu, Python Dianggap sebagai High Level Programming*. Retrieved from Universitas Multimedia Nusantara: https://www.umn.ac.id/pemula-wajib-tahu-python-dianggap-sebagai-high-level-programming/

Rusdi, J. F. (2022, Mei 3). *Computer Vision: Pemanfaatan Kecerdasan Melalui Mata-nya Komputer*. Retrieved from Kompasiana.com: https://www.kompasiana.com/inijack/62958ab4bb448663d1233902/computer-vision-pemanfaatan-kecerdasan-melalui-mata-nya-komputer?

Sidharta, H. A. (2017). *Introduction to Open CV*. Retrieved from Binus University: https://binus.ac.id/malang/2017/10/introduction-to-open-cv/