BAB II LANDASAN TEORI

A. Tinjauan Pustaka

1. Penelitian Terkait

Deteksi Kendaraan secara *real time* menggunakan metode YOLO (*You Only Look Once*), hasil dari penelitian ini adalah pendeteksian kendaraan dengan menerapkan metode YOLO (*You Only Look Once*) memiliki *performance* yang dapat mendeteksi kendaraan sesuai dengan varian kendaraan yang telah diuji yaitu sepeda motor, mobil, truk dan bus [7].

Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object *Tracking*, hasil dari penelitian ini yaitu: Sistem yang dibuat dapat mendeteksi dan menghitung Objek yang telah ditentukan, perbedaan warna objek dan latar belakang objek sangat menentukan keberhasilan pendeteksian objek, serta pencahayaan yang baik untuk pendeteksian objek adalah 31- 15.000 lux [1].

Deteksi Kendaraan Bergerak Secara Real Time, hasil dari penelitian ini yaitu: Hasil pendetekesian dalam sistem ini sangat dipengaruhi oleh pencahayaan, proses segmentasi dalam sistem ini sangant tergantung .dengan background yang diambil, semakin mirip dengan dengan foreground semakin baik, sehingga pendeteksian tipe menjadi semakin tepat [8].

Penelitian yang akan dilakukan, penulis menggunakan metode YOLO untuk mendeteksi objek seperti staff, dosen dan mahasiswa fakultas ilmu Penelitian terkait diatas memiliki kesamaan atau keterkaitan pada penelitian yang akan dilakukan. Pada penelitian terkait diatas terdapat penelitian atau jurnal terkait dengan penelitian yang akan dilakukan yaitu penelitian tentang penerapan metode YOLO (*You Only Look Once)*. Perbedaan penelitian yang akan dilakukan dengan penelitian terkait terletak pada objek penelitian. Objek dari penelitian terakait menerapkan metode YOLO

untuk mendeteksi kendaraan seperti motor, mobil, truk dan bus secara *real time*.

2. Landasan Teori

1. *Object Detection*

Object *Detection* mengidentifikasi kelas objek dengan melakukan training menggunakan objek pada database dan mengelompokan objek menjadi 2 kelas, yaitu kelas yang mewakili objek dan kelas yang mewakili non-objek. Object *Detection* selanjutnya dapat dibagi menjadi soft *Detection*, yang hanya mendeteksi keberadaan suatu objek, dan hard *Detection*, yang mendeteksi keberadaan dan lokasi objek [9].

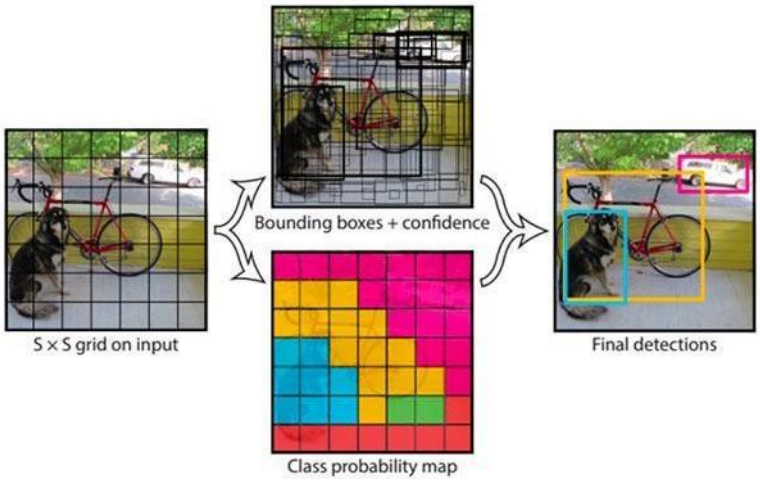
2. *Object Tracking*

Dalam visi komputer, object*Tracking* (pelacakan objek) adalah suatu proses untuk melacak satu objek atau lebih dari suatu citra. Object*Tracking* termasuk dalam salah satu fungsi yang sangat penting di bidang visi komputer. Ada tiga langkah penting dalam analisa video: deteksi objek yang bergerak, mendeteksi beberapa objek di setiap frame, dan analisa objek yang dilacak untuk mengenali pergerakaan objek pada citra.Dalam bentuk yang paling sederhana, *Tracking* dapat didefinisikan sebagai suatu masalah untuk memperkirakan lintasan dari sebuah objek yang bergerak dalam sebuah gambar. Secara konsisten, pelacak memberikan label pada objek yang dilacak pada frame-frame yang berbeda dalam sebuah video [10].

3. *You Only Look Once* (YOLO)

Algoritma YOLO adalah metode yang didasarkan dengan regresi yang memprediksi menggunakan *class object* dan *bounding box* untuk menentukan lokasi dari sebuah objek pada citra dalam satu kali algoritma. Algoritma YOLO bekerja dengan cara membagi citra menjadi beberapa sel, setiap sel digunakan untuk memprediksi 5 *bounding box* jika objek pada citra ada lebih dari satu. Prediksi akan menghasilkan nilai prediksi. Hasil prediksi akan dikumpulkan dan *bounding box* dengan probabilitas terkecil akan dihapus. *Bounding box*

dengan nilai probabilitas prediksi terbesar akan menjadi hasil akhir (Maj, Michal, 2018).



Gambar 1. Ilustrasi Algoritma YOLO

(Sumber :https://www.pyimagesearch.com/2018/11/12/yolo-object-Detection-with- opencv/)

Vektor Prediksi

Prediksi pada YOLO menggunakan arsitektur yang mirip seperti *Convolutional Neural Network* (CNN). YOLO hanya menggunakan *convolutional layer* dan *pooling layer*. *Convolutional layer* terakhir pada arsitektur YOLO disesuaikan dengan jumlah kelas dan jumlah kotak prediksi yang ditentukan. Rumus untuk menghitung ukuran keluaran pada *convolutional layer* terakhir dapat dilihat pada persamaan (1).

$$Y = S, S, B \times (5 + C) \quad (1)$$

Keterangan :

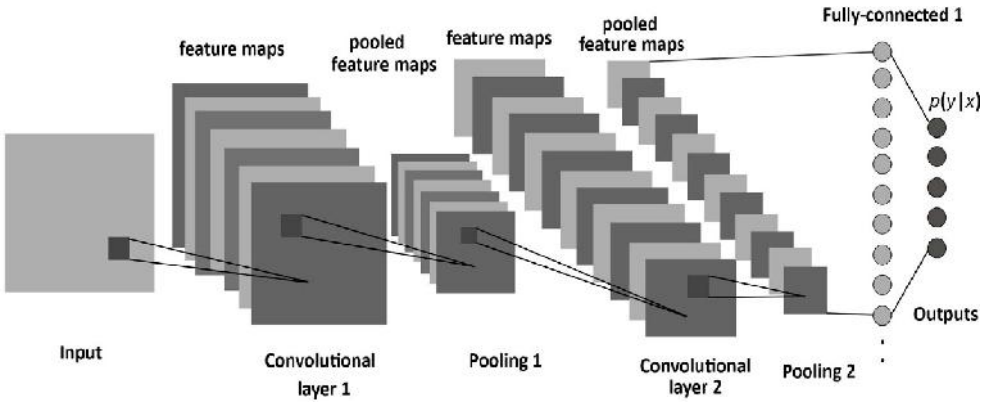
Y = Ukuran Keluaran

S = Jumlah baris atau kolom grid

B = Jumlah prediksi yang diinginkan pada setiap grid

C = Jumlah kelas yang ingin diprediksi

Sebelum pendeteksian, algoritma YOLO membutuhkan proses anotasi terlebih dahulu untuk data. Setiap data memiliki nama kelas, titik koordinat X objek, titik koodinat Y objek, panjang *bounding box*, dan lebar *bounding box*.

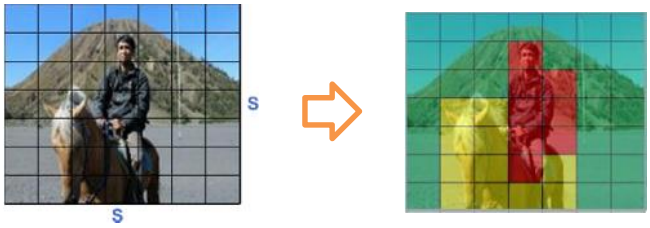


Gambar 2. Arsitektur Convolution Neural Network

(Sumber : https://www.mdpi.com/1099-4300/19/6/242)

Klasifikasi secara umum adalah proses untuk mengidentifikasi label dari data yang diuji. YOLO menerapkan *localization* pada pengklasifikasiannya, yaitu terdapat tambhan pemberian lokasi objek dalam bentuk *bounding box* (b_x,b_y,b_h,b_w) [11], seperti pada gambar 1. Tahapan algoritma YOLO adalah sebagai berikut.

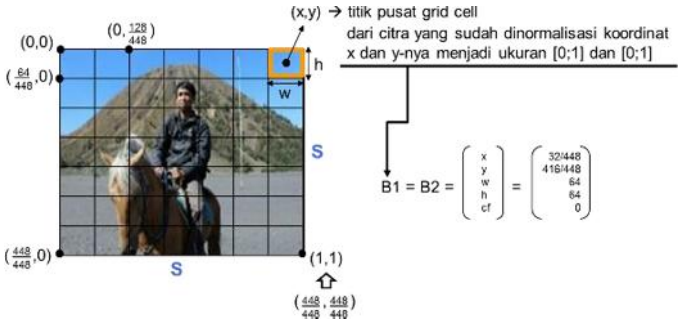
1. Baca citra dengan ukuran sembarang.
2. Ubah Ukuran citra menjadi 448 x 448, lalu buat *grid* pada citra dengan ukuran *S* x *S grids.*



Gambar 3. Grid Algoritma YOLO
(sumber: [11])

Jika *S*=7, maka tiap *grid cell* ukurannya 64 x 64. Sehingga terdapat sebanyak 49 *grid cell*. Dan misalkan pada citra ada 2 kelas (*nC*=2), yaitu “Manusia” (*c1*), “Kuda” (*c2*), dan “Background”.

3. Tiap *grid cell*, missal *nB*=2, terdapat *B* yang berisi 5 nilai, yotu lokasi kordinat *x* diasumsikan berdasa baris (*x_br*), kordinat *y* berdasarkan kolom (*y_kol*), ukuran nilai *confidence* (*x,y,w,h,cf*) terhdapat *nB* bbox yang ada, yaitu B1 dan B2.

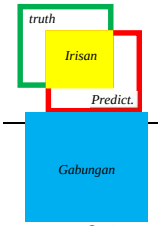


Gambar 4 *Grid algoritma dengan nilai* confidence

(sumber : [11])

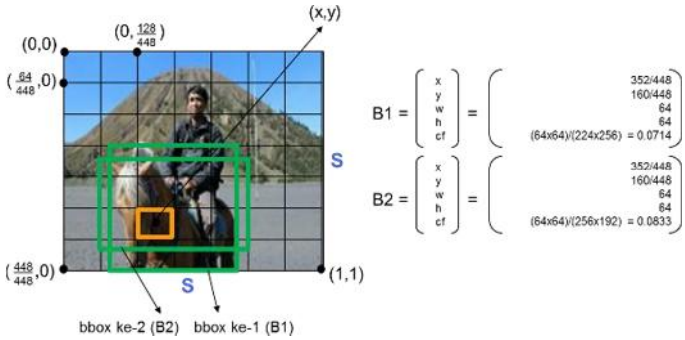
cf = 0 (selalu di-*set* = 0, jika dalam *grid cell* adalah *background*). *Confidence* (*cf*) = P(object)*IoU, yang mana, *nB* menyatakan banyaknya *bbox*, ***B1*** untuk *bbox1*, *B2* untuk *bbox2*, dan *bbox* menyatakan *bounding box*. Persamaan 2 merupakan *Intersection Over Union (IoU).*

$$IoU = \frac{I_1}{G} \qquad (3)$$



Pada perhitungan *confidence (cf)* = p(*object*)*IoU, p(*object*) dapat dilewati terlebih dahulu, sehingga cukup dengan *cf=IoU.* Karena P($class_i$ | *object*)* P(*object*)*IoU= P($class_i$)*IoU.

4. Jika 2 bbox mengacu pada kelas yang sama, maka hail ukuran tensornya adalah (*S* x *S* x (*nB* x 5 + *nC*)) = (7 x 7 x (2 x 5 + 2)) = (7 x 7 x 12) tensor.

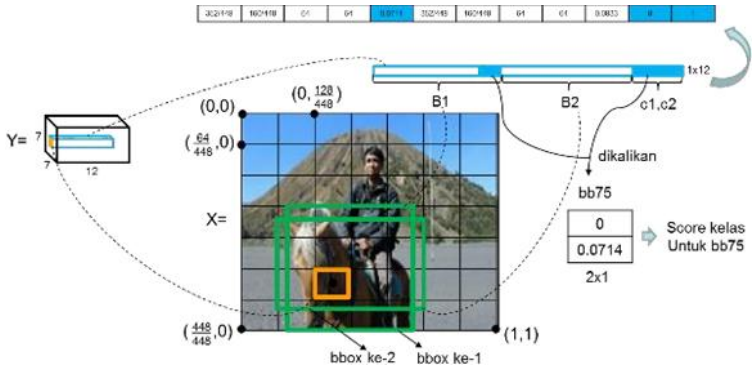


Gambar 5. Grid algoritma acuan kelas

(sumber : [11])

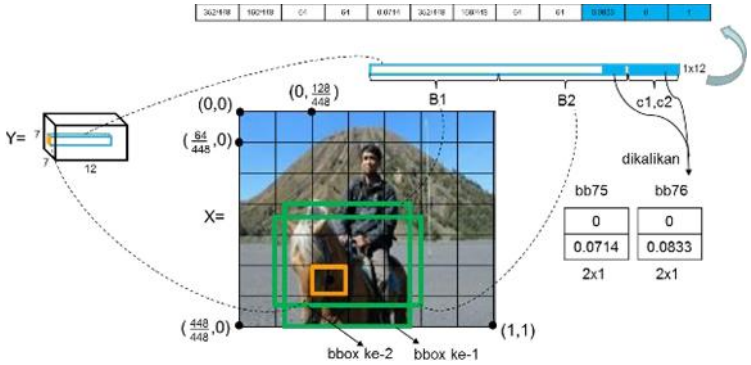
Dimana, bbox ke-1 dan bbox ke-2, dibuat bebas dan sebaiknya berbeda. Hitungan matriks *bb* dengan ukuran (*nCx(SxSxnB)*) = 2 *x (7x7x2) = (2 x98)*, mulai dari *grid cell*

ke-1 sampai ke-49 pada 2 bbox yang mengacu pada kelas yang sama, missal untuk bb75 dan bb76.



Gambar 6. Grid algoritma bb75

(sumber : [11])



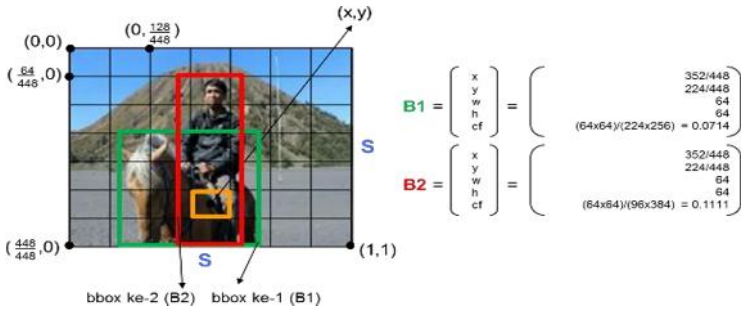
Gambar 7. Grid algoritma bb76

Hasil dari matriks bb.

Gambar 8. Hasil matriks

(sumber : [11])

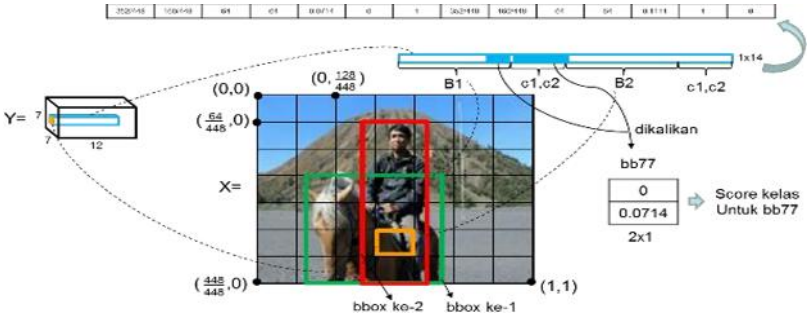
5. Jika 2 bbox mengacu pada kelas yang berbeda atau karena terdapat *overlapping object*, maka hasil ukuran tensornya adalah (S x S x (nB x 5 + 2 x nC)) = (7 x 7 x (2 x 5 + 2 x 2)) = (7 x 7 x 14) tensor.



Gambar 9. *Grid algoritma* overlapping object

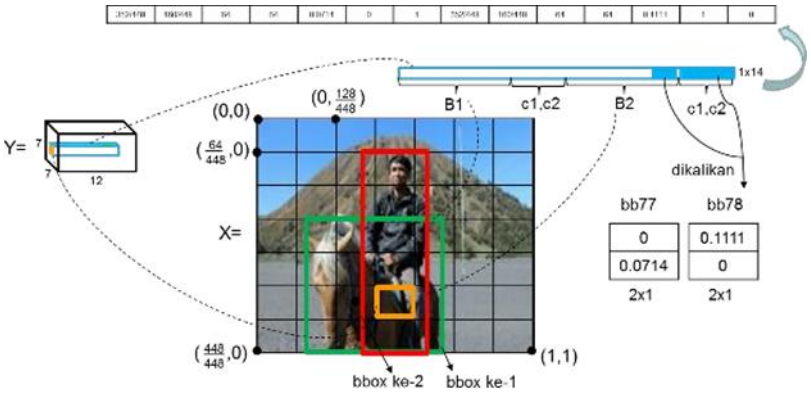
(sumber : [11])

Dimana, bbox ke-1 dan bbox ke-2, mengacu pada objek yang berbeda kelas. Hitung matriks bb dengan ukuran (*nC* x (*S* x *S* x *nB*)) = 2 x (7 x 7 x 2) = (2 x 98), mulai dari *grid cell* ke-1 sampai ke-49 pada 2 *bbox* yang mengacu pada kelas yang berbeda, misal untuk bb77 dan bb78.



Gambar 10. *Grid algoritma bb77*

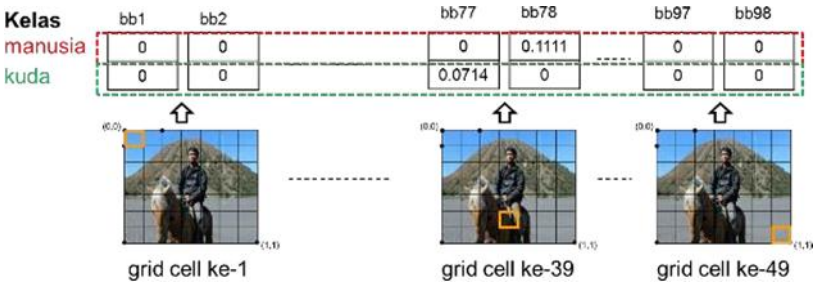
(sumber : [11])



Gambar 11. Grid algoritma bb78

(sumber : [11])

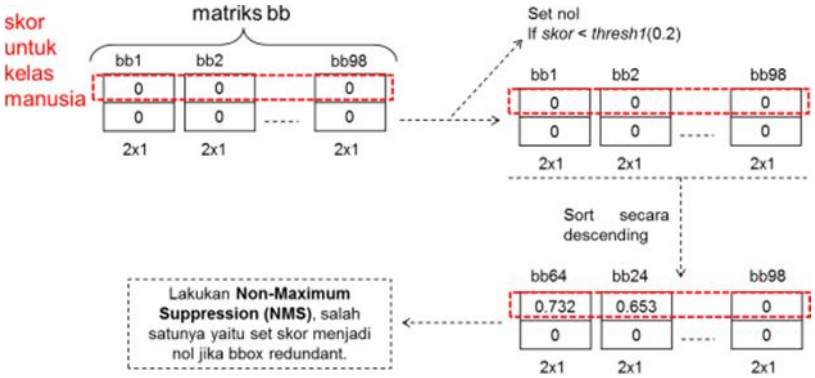
Hasil dari matriks bb.



Gambar 12. Hasil matriks grid bb kuda dan manusia

(sumber : [11])

6. Jika kelas pada matriks bb, lakukan set skor = 0, jika skor < *thresh(0.02)*, kemudia urutkan sacara descending.



Gambar 13. *Urutan matriks* descending

(sumber : [11])

7. Lakukan *Non-Maximum Suppression (*NMS*).*

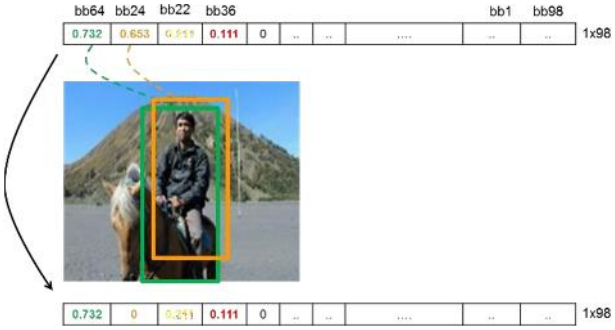
- Daftar semua bbox

Gambar 14. Grid algoritma non-maximum suppression

(sumber : [11])

- Set bbox dengan skor maks, sebagai “bbox_max”. Misal bb64 deketahui sebagai bbox_max.

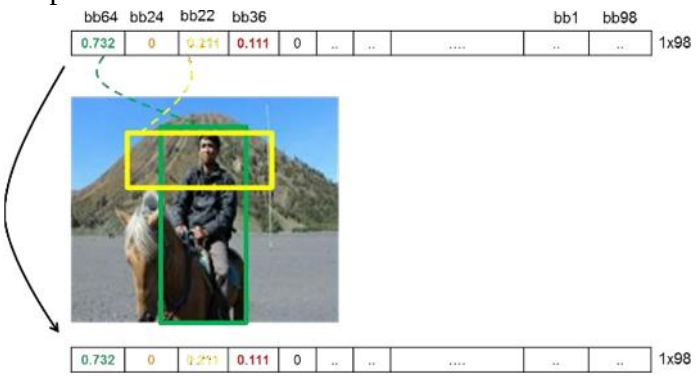
- Bandingkan “bbox_max” dengan bbox lainnya seagai “bbox_cur” yang memiliki skor dibawahny dan tidak nol. Jika *IoU*(bbox_max,bbox_cur) > 0.5, maka set skor nol untuk bbox_cur. Dari hasil perbandingan, jika bbox_max = bb64, bbox_cur = bb24, maka IoU(bbox_max,bbox_cur) > 0.5 (true), maka set skor bb24 = 0.



Gambar 15. *Grid hasil perbandingan* true

(sumber : [11])

- Lanjut ke bb22, dari hasil perbandingan, jika bbox_max = bb64, bbox_cur=bb22, maka IoU(bbox_max,bbox_cur) > 0.5 (false), maka skor bb22 tetap.



Gambar 16. Grid algoritma hasil perbandingan false bb22

(sumber : [11])

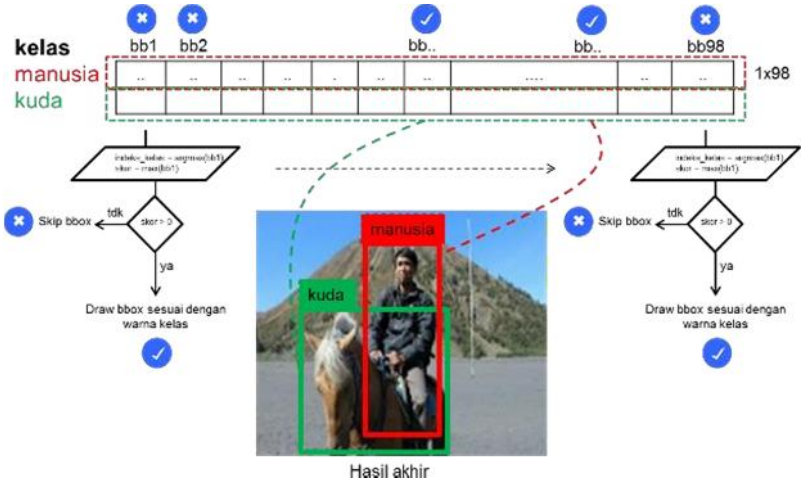
- Lanjutkan ke bb36, dari hasil perbandingan, jika bbox_max = bb64, bbox_cur = bb36, maka IoU(bbox_max,bbox_cur) > 0.5 (false), maka skor bb36 tetap.

Gambar 17. Grid algoritma hasil perbandingan false bb36

(sumber : [11])

- Lanjutkan melakukan pembandingan sampai bbox_max = bb64, bbox_cur = bb98. Lalu lanjutkan lagi, set bbox_max = bb22, dan bbox_cur = bb36, dst.
- Kemudian lanjutkan ke kelas berikutnya, lakukan hal yang sama seperti yang telah dilakukan pada kelas manusia

8. Plot *bounding box* berdasarkan hasil NMS



Gambar 18. Hasil grid algoritma YOLO berdasarkan NMS

(sumber : [11])

4. *Computer Vision*

Computer Vision merupakan salah satu cabang dari kecerdasan buatan (artificial intelligence) yang difokuskan pada pengembangan algoritma untuk menganalisis informasi dari suatu citra ke dalam bentuk informasi yang sebenarnya di dunia nyata. Saat ini deep

learning banyak digunakan pada penelitian mengenai *Computer Vision* [12].

Gambar 19. *Hasil dari* computer vision

(sumber: https://miro.medium.com/)

Saat seorang mengendarai mobil dan kemudian melihat ada objek yang tiba – tiba bergerak ke dalam jalur mobil, maka pengemudi harus bereaksi langsung seperti menginjak rem (mengerem). Maka pengemudi tersebut telah melakuakan tugas yang sangat kompleks, yaitu mengidentifikasi objek, memproses dan kemudian memutuskan tindakan yang akan diambil. Tujuan *computer vision* adalah memungkinkan sebuah komputer untuk melakukan tugas yang sama dengan yang dilakukan pengemudi dengan efisien yang sama.

5. *OpenCV*

OpenCV (*Open Source Computer Vision*) adalah *library* dari fungsi pemrograman untuk *realtime computer vision* [1]. *OpenCV* menggunakan lisensi BSD dan bersifat gratis baik untuk penggunaan akademis maupun komersial. *OpenCV* dapat digunakan dalam bahasa pemrograman C, C++, Python, Java, dan sebagainya [2]. *OpenCV* dapat digunakan pada sistem operasi Windows, Linux, Android, iOS dan Mac OS. *OpenCV* memiliki lebih dari 2500 algoritma yang telah dioptimalkan [13].

6. *Phyton*

Bahasa pemrograman *Python* merupa- kan bahasa pemrograman popular yang memiliki keunggulan sebagai berikut :

1. Mudah untuk digunakan dalam mengembangkan sebuah produk perangkatlunak, perangkat keras, *Internet of Things*, aplikasi web, maupun video game.
2. Selain memiliki keterbacaan kode yang tinggi, sehingga kode mudah dipahami, bahasa pemrograman ini memiliki *library* yang sangat banyak dan luas.
3. Merupakan bahasa yang mendukung ekosistem *Internet of Things* dengan sangat baik Internet [14].

7. *Image Processing*

Dengan kemajuan ilmu teknologi pengolahan citra digital (*Digital Image Processing*) yang semakin pesat, maka dapat mempermudah kehidupan manusia, dan dewasa ini banyak aplikasi yang dapat menerapkannya, dalam berbagai bidang. Pengolahan citra (*image processing*) adalah teknik mengolah citra yang mentransformasikan citra masukan menjadi citra lain agar keluaran memiliki kualitas yang lebih baik dibandingkan kualitas citra masukan. Pengolahan citra sangat bermanfaat, diantaranya adalah untuk meningkatkan kualitas citra, menghilangkan cacat pada citra, mengidentifikasi objek, penggabungan dengan bagian citra yang lain. Dengan memanfaatkan teknologi tersebut, maka diharapkan adanya suatu aplikasi yang dapat menangkap suatu obyek yang ada di depan kamera bisa mengidentifikasi jenis objek serta melakukan *Tracking* objek secara real-time [15].

8. *Google Colaboratory*

Google Colaboratory atau *Google Colab* merupakan tools yang berbasis *cloud* dan *free* untuk tujuan penelitian. *Google Colab* dibuat dengan *envirotment* jupyter dan mendukung hamper semua *liblary* yang dibutuhkan dalam lingkungan pengembangan *Artificial Intelegence* (AI). Penggunaan *Google Colab* ditunjukan bagi para peneliti yang sedang mengembangkan penelitian dan membutuhkan spesifikasi komputer yang tinggi [16]. *Google Colab* bisa dilakukan untuk:

1. Meningkatkan keterampilan bahasa pemrograman python.
2. Mengembangkan aplikasi seperti keras, tensorflow, pytorch dan OpenCv.

Fitur *Google Colab* yang paling penting dan yang membedakan *Colab* dengan yang lain *cloud* gratis lainnya adalah *Colab* benar – benar menyediakan GPU sebesar 12 GB secara gratis.

9. *Webcam*

Webcam merupakan sebuah kamera yang terhubung dengan komputer/laptop. *Webcam* menangkap gambar diam maupun bergerak atau vidio. Dengan bantuan peranhkat lunak, *webcam* dapat mengirimkan video ke internet secara *real-time* [17].

Webcam tidak seperti kamera digital dan juga *camrecorder* digital yang memiliki penyimpanan tersendiri. Maka dari itu *webcam* harus selalu terhubung ke komputer dan menggunakan penyimpanan komputer untuk menyimpan gambar maupun video.

10. *Artificial Intelligence* (AI)

Kecerdasan Buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. Menurut John McCarthy, AI : untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman, penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik [18].

B. Kerangka Pikir

Judul

Object Detection dan Tracking menggunakan algoritma *You Only Look Once* berbasis *Computer Vision*

↓

Masalah Utama

Salah satu bidang ilmu yang mendukung dalam mempermudah pekerjaan manusia ialah *Computer Vision.* Dalam beberapa tahun terakhir, dengan pesatnya perkembangan *Computer Vision*, *Objeck Detection* sebagai bagian penting dari *Computer Vision* telah banyak digunakan di berbagai bidang. *Objeck Detection* berdasarkan pemrosesan gambar mengekstrak fitur dari gambar, lalu memperoleh dan menganalisis informasi objek seperti kategori, posisi dan arah.

↓

Teori Penyelesaian Masalah

Metode yang digunakan dalam mengembangkan sistem informasi penggajian ini adalah dengan menggunakan Algoritma *You Only Look Once* (YOLO) untuk melakukan deteksi dan *tracking* objek secara *real-time* berdasarkan *Convolutional Neural Network (CNN).* Algoritma ini akan membagi suatu citra ke dalam 13x13 bagian yang di dalamnya terdapat *Bounding Boxes.*

↓

Solusi Dari Masalah Berdasarkan Teori

Keuntungan menggunakan algoritma YOLO adalah proses pengembangan yang di kembangkan dengan keungulan komputasi yang lebih ringan dari algoritma lainnya dengan kecepatan deteksi 22 *ms*/gambar dengan 45fps (*frame per second).*

↓

Hasil Dari Penelitian

Aplikasi berbasis Python yang dapat mendeteksi dan mentracking suatu objek berdasarkan class yang sudah dibuat secara real-time dengan menggunakan webcam

BAB III. METODOLOGI PENELITIAN

A. Tahapan Penelitian

Kegiatan yang dilakukan dalam penelitian diantaranya adalah:

a. Mengidentifikasi Masalah

   Tahap ini, peneliti harus terlebih dahulu mencari apa saja masalah yang hendak diteliti. Masalah yang berhasil diidentifikasi pada penelitian ini adalah bagaimana sebuah kamera mampu mendeteksi dan melacak suatu objek menggunakan algoritma YOLO berbasis *Computer Vision* secara statis maupun *real-time*.

b. Merumuskan Masalah

   Tahap ini merupakan penemuan masalah yang kemudian peneliti harus membuat rumusan masalah berdasarkan masalah – masalah yang akan diteliti. Termasuk batasan – batasan masalah dalam mendeteksi dan mengenali suatu objek menggunakan algoritma YOLO. Hal ini sangat penting untuk menghindari pembahasan yang tidak berkaitan dengan masalah yang telah dirumuskan sebelumnya.

c. Mengadakan Studi Pendahuluan

   Hal ini dilakukan dengan tujuan untuk mengumpulkan informasi – informasi yang berkaitann dengan masalah yang akan diteliti. Penulis melakukan studi pendahuluan secara teoritis melalui jurnal, skripsi dan buku yang berakitan dengan penelitian *object Detection* dan *Tracking* menggunakan algoritma YOLO berbasis *Computer Vision.*

d. Menentukan Sampel Penelitian

   Pada tahap ini, peneliti menentukan objek yang akan diteliti. Objek yang akan diteliti yaitu staff, dosen dan mahasiswa fakultas ilmu komputer. Data gambar yang telah dikumpulkan akan diintegrasikan kedalam dataset. Data berupa gambar format *.jpg*. Dataset yang diperoleh akan dibagi menjadi *training set* dan *testing set. Training set* adalah bagian dataset yang dilatih untuk membuat prediksi atau menjalankan fungsi dari algoritma YOLO yang diberi petunjuk agar data yang dilatih bisa mencari korelasinya sendiri. *Testing set* adalah bagian dataset yang dites untuk melihat keakuratannya dari *training set.*

e. Tahap Rancangan Sistem

   Ditahap ini peniliti akan merancang sebuah sistem mendeteksi dan melacak objek menggunakan algoritma YOLO berbasis *computer vision*.

f. Uji Coba Program

Pada tahap ini peniliti akan melakukan uji coba program guna mengetahui kesalahan yang mungkin terjadi pada program/aplikasi.

g. Implementasi Program

Pada tahap ini rancangan diimplementasikan kedalam bentuk kode – kode program *python* sehingga dapat mendeteksi dan melacak objek menggunakan algoritma YOLO.

h. Pembimbingan Penulisan Naskah Skripsi

Pada tahap ini di lakukan aktivitas bimbingan proposal untuk mendapatkan saran dan masukan yang sesuai dengan Panduan pengerjaan proposal Fakultas Ilmu Komputer.

i. Pendadaran

Ujian yang ditujukan untuk mengevaluasi proposal dan menguji pengetahuan sehubungan hasil riset dan pengetahuan umum yang terkait

Tabel 1. Jadwal Penelitian

No	Tahap Penelitian	September 2021		Oktober 2021				November 2021				Desember 2021				Januari 2022		
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3
1.	Identifikasi Masalah																	
2.	Merumuskan Masalah																	
3.	Mengadakan Studi Pendahuluan																	
4.	Menentukan Sampel Penelitian																	
5.	Tahap Rancangan Sistem																	
6.	Uji Coba Program																	
7.	Implementasi Program																	
8.	Pembimbingan Penulisan Naskah Skripsi																	
9.	Pendadaran																	

B. Instrumen Penelitian

Instrumen penelitian adalah alat bantu yang dipilih dalam kegiatannya agar sistematis dan mempermudah peneliti selama melakukan penelitian. Instrumen ini terbagi menjadi dua yaitu:

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan adalah:

1. Laptop
2. *Processor* Core ™ i5-8250U.
3. 4 GB DDR 4 Memory
4. NVDIA® GeForce® MX130 with 2 GB VRAM
5. Webcam Logitech Brio 4K

b. Perangkat Lunak (*Software*)

Perangkat lunak atau *software* yang digunakan adalah:

1. Microsoft Windows 10 Professional 64-bit, sebagai Sistem Operasi.
2. PyCharm 2021.1 (Community Edition)
3. Goggle Colab
4. LabelImg

C. Lokasi Penelitian

Lokasi penelitian bertempat di Fakultas Ilmu Komputer, Universitas Muslim Indonesia, Jalan Urip Sumoharjo Km. 5 Makassar.

D. Metode Penelitian

a. Studi Literatur dan Diskusi

Merupakan metode yang dilakukan oleh penulis dengan mengumpulkan referensi seperti jurnal, dan skripsi tentang penelitian *object detetction* dan *Tracking* menggunakan algoritma YOLO berbasis *Computer Vision.* Serta diskusi dengan dosen pembimbing tentang penelitian mendeteksi dan melacak objek berbasis *Computer Vision.*

b. Analisis Sistem

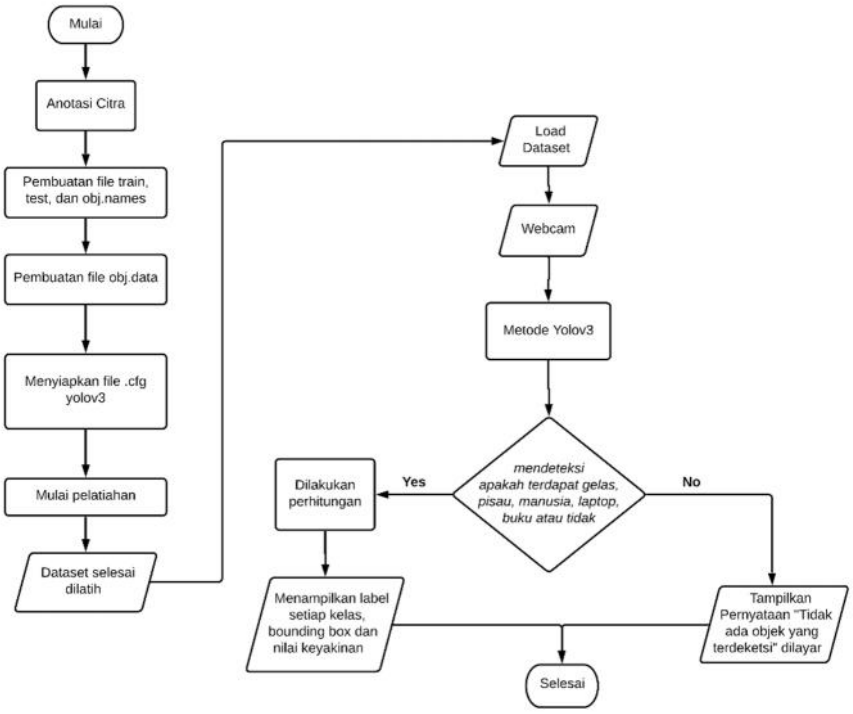
Pada tahap ini untuk sistem ususlan penulis memberikan solusi untuk mendeteksi dan melacak suatu objek pada kamera, yaitu merancang sebuah program *object detetction* dan *Tracking* menggunakan algoritma YOLO berbasis *Computer Vision.* Program dibuat menggunakan bahasa pemrograman *python*. Program ini mampu mendeteksi dan melacak suatu objek secara statis maupun dinamis (*real-time*). Adapun sistem berjalan saat ini yaitu pemantauan atau pengawasan dari kamera hanya menampilkan objek saja tanpa adanya pengenalan objek. Tujuan

pembuatan program ini untuk memudahkan manusia dalam hal pengawasan atau pemantauan.

c. Perancangan Sistem

a) Flowchart Sistem *Detection* dan *Tracking object*

Flowchart adalah suatu symbol tentang urutan proses suatau sistem mulai dari pengolahan data serta beberapa prosedur pemecahan masalah dari suatu aplikasi dan akan menampilkan nilai dari output sistem.



Gambar 20. Flowchart Sistem Detection *dan* Tracking object

Flowchart dirancang agar dapat mengambarkan secara garis besar proses kerja sistem dalam melakukan pendeteksian. Dapat dilihat pada gambar 20. *flowchart* alur kerja sistem terbagi menjadi dua tahap yaitu tahap pertama bisa disebut dengan proses tahap *training* dan tahap kedua adalah tahap pengimplementasian model.

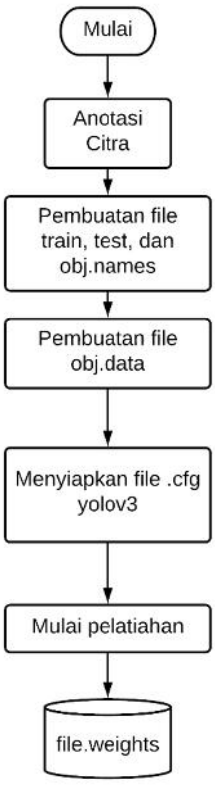
Tahapan *training* adalah tahapan awal yang mulai dari mempersiapkan data latih yang sudah diberi label menggunakan aplikasi LabelImg untuk selanjutnya dimasukan pada proses *training dataset*.

Pada tahapan implementasi model, setelah menyelesaikan proses *training* maka model yang didapatkan dapat dijalankan menggunakan metode YOLO ketika diberikan inputan yang berasal dari *webcam* yang merupakan alat yang digunakan sebagai

pengambil gambar yang nantinya akan didteksi. Hasil dari pengambilan gambar diproses dengan menggunakan metode YOLO dengan menggunakan *dataset* yang sudah dilatih. Terdapat dua percabangan, yang pertama apabila terdapat objek gelas, pisau, manusia, laptop, atau gelas dan yang kedua apabila tidak terdapat objek gelas, pisau, manusia, laptop, atau gelas. Apabila terdapat objek yang tedeteksi, maka proses selanjutnya adalah menghitung jumlah objek yang terdeteksi, setelah proses perhitungan selesai maka hasil yang sudah didapatkan nantinya akan ditampilka dilayar. Namun apabila objek tidak terdeteksi maka akan langsung muncul pemberitahuan objek tidak terdeteksi.

b) *Training Dataset*

Gambar 21 menunjukan langkah – langkah *training dataset* YOLO menggunakan *Google Colab.*



Gambar 21. *Flowchart Training Dataset*

Berdasarkan gambar diatas dijelaskan bahwa:

1. Hal pertama yang harus disiapkan adalah gambar – gambar obejek yang akan diteliti. Dalam penelitian ini, objek yang akan dideteksi adalah staff, dosen dan mahasiswa fakultas ilmu komputer. Jadi, pengambilan gambar akan diambil dengan berbagai ukuran dan juga dari berbagai posisi baik dari atas, samping dan sebagainya.
2. Setelah gambar objek yang akan digunakan sudah siap, maka dilakuakan anotasi citra. Anotasi citra merupakan proses membuat label dengan memberikan kotak pembatas (*bounding boxes*) beserta nama kelas pada setiap citra. Aplikasi yang digunakan dalam proses ini adalah labelImg.
3. Setelah anotasi citra dilakukan, sebelum melakukan training *dataset* terlebih membuat file *train, test* dan juga *.names*. file *train* dan juga *test* merupakan file yang berisi alamat gambar yang nantinya akan digunakan untuk bahan *training dataset* dan juga file *.names* merupakan nama kelas yang akan digunakan untuk *training dataset*.
4. Setelah semua file diatas selesai dibuat, maka selanjutnya adalah membuat file dengan ekstensi *.data*. File ini merupakan file yang digunakan untuk alamat dari ketiga file yang dibuat diatas.
5. Meyiapkan file *cfg*, file ini sangat penting dalam proses training dataset. File *cfg* merupakan kode untuk model.
6. Setelah semua yang dibutuhkan sudah disiapkan maka dilanjutkan dengan mulai pelatihan. Untuk peltihan dilakukan sebanyak 2000 iterasi atau perulangan dikarenakan *maks_batch* yang diguanakan adalah 2000. Didalam proses pelatihan nantinya sistem akan melakukan penyimpanan otomatis setiap 100 dan juga 1000 iterasi. Hal ini dilakukan karena proses pelatihan menggunakan internet. Jadi apabila koneksi terputus maka nantinya sistem akan melanjutkan proses pelatihan dari nilai iterasi terakhir tersimpan. Hal ini dilakukan sampai pada iterasi ke 2000 atau selesai. Setelah proses pelatihan selesai maka file bobot atau dataset bisa digunakan.
7. Setelah *training* selesai, maka hasil dari *training dataset* akan tersimpan kedalam *database* dengan nama file.weights.

c) *Testing Dataset*

Gambar 22 menunjukan langkah – langkah *testing dataset* YOLO menggunakan *Google Colab.*

Gambar 22. *Flowchart Testing Dataset*

Berdasarkan gambar diatas dijelaskan bahwa:

1. Hal pertama yang harus disiapkan adalah gambar – gambar obejek yang akan diteliti.
2. Setelah gambar objek yang akan digunakan sudah siap, maka dilakuakan anotasi citra. Anotasi citra merupakan proses membuat label dengan memberikan kotak pembatas (*bounding boxes*) beserta nama kelas pada setiap citra. Aplikasi yang digunakan dalam proses ini adalah labelImg.
3. Setelah anotasi citra dilakukan, sebelum melakukan training *dataset* terlebih membuat file *train, test* dan juga *.names*. file *train* dan juga *test* merupakan file yang berisi alamat gambar yang nantinya akan digunakan untuk bahan *training dataset* dan juga file *.names* merupakan nama kelas yang akan digunakan untuk *training dataset*.
4. Setelah semua file diatas selesai dibuat, maka selanjutnya adalah membuat file dengan ekstensi *.data*. File ini merupakan

file yang digunakan untuk alamat dari ketiga file yang dibuat diatas.

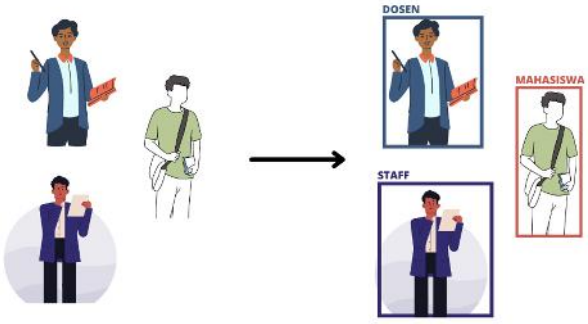
5. Meyiapkan file *cfg*, file ini sangat penting dalam proses training dataset. File *cfg* merupakan kode untuk model.
6. Setelah semua yang dibutuhkan sudah disiapkan maka dilanjutkan dengan mulai pelatihan. Untuk peltihan dilakukan sebanyak 2000 iterasi atau perulangan. Hal ini dilakukan sampai pada iterasi ke 2000 atau selesai. Setelah proses pelatihan selesai maka file bobot atau dataset bisa digunakan.
7. Setelah *training* selesai, maka hasil dari *training dataset* akan tersimpan kedalam *database* dengan nama file.weights.
8. Lakukan *testing* klasifikasi dengan menggunakan algoritma YOLOv3.
9. Tampilkan nilai akurasi yang didapatkan.

d) Arsitektur Sistem

Gambar 23. Arsitektur Sistem

e) Cara Kerja Sistem

1. Gunakan webcam untuk pengambilan gambar.
2. Hubungkan webcam ke komputer.
3. Ketika sudah terhubung maka jalankan program pendeteksi dan pelacakan objek.
4. Arahkan webcam ke objek yang akan di deteksi (staff, dosen dan mahasiswa fakultas ilmu komputer).
5. Setelah itu tunggu dan hasil akan keluar, objek sudah berhasil dideteksi dan dilacak.



Gambar 24. Proses deteksi objek