SISTEM DETEKSI OBJEK MANUSIA MENGGUNAKAN ALGORITMA YOLOV8 BERBASIS KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)

SKRIPSI

Disusun Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik
Pada Program Studi Teknik Informatika Universitas Sangga Buana YPKP

Disusun oleh:

FAHRI MUHAMAD ZULKARNAEN

2113191049

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS SANGGA BUANA YPKP

2024

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Fahri Muhamad Zulkarnaen

NIM : 2113191049

Program Studi : Teknik Informatika

Menyatakan dengan sebenar-benarnya bahwa laporan Skripsi saya yang berjudul:

SISTEM DETEKSI MANUSIA MENGGUNAKAN ALGORITMA YOLOV8 BERBASIS KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)

Adalah hasil karya sendiri dan bukan jiplakan hasil karya orang lain.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Jika dikemudian hari terbukti bahwa laporan Skripsi saya merupakan hasil jiplakan, maka saya bersedia menerima sanksi sesuai dengan peraturan perundang-undangan yang berlaku.

Bandung, 5 Maret 2024

Fahri Muhamad Zulkarnaen

LEMBAR PERSETUJUAN DAN PENGESAHAN TUGAS AKHIR

Skripsi ini diajukan oleh:

Nama	:	Fahri Muhamad Zulkarnaen
NPM	:	2113191049
Program Studi	:	Teknik Informatika
Judul	:	SISTEM DETEKSI MANUSIA MENGGUNAKAN ALGORITMA YOLOV8 BERBASIS KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)

Untuk dipertahankan sidang Skripsi Semester Ganjil tahun 2024 di hadapan para penguji dan diterima sebagai bagian dari persyaratan yang di perlukan untuk memperoleh gelar Sarjana Teknik (ST) pada Fakultas Teknik Program Studi S1 Teknik Informatika Universitas Sangga Buana YPKP.

Bandung, 05 Maret 2024

Menyetujui,

Pembimbing

Bambang Sugiarto.ST., M.T
NIDN: 8861060017

Penguji I

Slamet Risnanto, S.T., M.Kom.
NIDN. 0424047307

Penguji II

Gunawansyah, S.T., M.Kom.
NIDN. 0420027907

Mengetahui:

Ketua Program Studi S1 Teknik Informatika

Gunawan, ST., M.Kom., MOS., MTA., MCE
NIDN:040427604

ABSTRAK

Keamanan menjadi prioritas utama di lingkungan seperti perusahaan dan fasilitas terbatas lainnya, mendorong pengembangan sistem keamanan yang lebih canggih. Perkembangan dalam teknologi kecerdasan buatan (AI) memungkinkan pengembangan sistem keamanan dalam bentuk pendeteksian objek manusia. Dukungan dari teknologi kamera depth sensor membuka peluang untuk meningkatkan akurasi deteksi. Penelitian ini bertujuan untuk mengembangkan sistem deteksi objek manusia menggunakan algoritma YOLOv8 dan kamera depth sensor, dengan studi kasus pada CV. Ateri Global Teknologi. Pengujian dilakukan menggunakan dataset uji dalam bentuk lima format video. Rata-rata hasil pengujian dari semua video tersebut adalah Precision 97%, Recall 95%, F-Measure 96%, dan Accuracy 92%. Kesimpulan dari penelitian ini adalah bahwa sistem memiliki akurasi sebesar 92%, yang dipengaruhi oleh berbagai faktor seperti sudut kamera pada saat perekaman dataset.

Kata Kunci: Objek Manusia, YOLO, YOLOv8, Kamera Depth Sensor

ABSTRACT

Security is becoming a top priority in environments such as companies and other restricted facilities, encouraging the development of more sophisticated security systems. Developments in artificial intelligence (AI) technology enable the development of security systems in the form of human object detection. Support from kamera depth sensor technology opens up opportunities to increase detection accuracy. This research aims to develop a human object detection system using the YOLOv8 algorithm and a kamera depth sensor, with a case study on CV. Ateri Global Technology. Testing was carried out using a test dataset in the form of five video formats. The average test results from the videos are Precision 97%, Recall 95%, F-Measure 96%, and Accuracy 92%. The conclusion of this research is that the system has an accuracy of 92%, which is influenced by various factors such as the camera angle when recording the dataset.

***Keywords**: Human Object, YOLO, YOLOv8, Depth Sensor Camera*

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Allah SWT, yang dengan rahmat dan karunia-Nya skripsi yang berjudul "SISTEM DETEKSI MANUSIA MENGGUNAKAN ALGORITMA YOLOV8 BERBASIS KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)" ini dapat diselesaikan tepat pada waktunya. Skripsi ini ditulis dalam rangka memenuhi syarat untuk menyelesaikan Program Sarjana (S1) Fakultas Teknik jurusan Teknik Informatika di Universitas Sangga Buana YPKP.

Dalam penulisan skripsi ini, sangat penulis sadari bahwa akan sangat sulit untuk penulis menyelesaikannya tanpa bantuan, bimbingan serta dukungan dari berbagai pihak baik secara langsung maupun secara tidak langsung. Untuk itu penulis menyampaikan terima kasih yang tak terhingga kepada:

1. Bapak Dr. Didin Saepudin, S.E., M.Si selaku Rektor Universitas Sangga Buana YPKP Bandung.
2. Bapak Gunawan, S.T., S. Kom., MOS., MCE. selaku Ketua Program Studi S1 Teknik Informatika.
3. Bapak Bambang Sugiarto, S.T., M.T selaku dosen pembimbing yang sudah berkenan menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
4. Seluruh Dosen dan Staf Fakultas Teknik Universitas Sangga Buana YPKP Bandung.
5. Kedua orang tua penulis, yang selalu memberikan doa dan usaha terbaiknya untuk penulis.
6. Semua rekan-rekan seperjuangan Universitas Sangga Buana YPKP Bandung, yang selalu menyemangati penulis untuk menyelesaikan skripsi ini.
7. Semua pihak yang selalu menyempatkan waktunya serta selalu mencoba membantu penulis menjadi pribadi yang lebih baik.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi saat ini sedang mengalami kemajuan pesat, salah satunya adalah dalam bidang teknologi kecerdasan buatan atau yang sering disebut sebagai *Artificial Intelligence* (AI). AI adalah representasi dari kemampuan manusia yang dimodelkan ke dalam sebuah mesin, yang kemudian diprogram untuk dapat berpikir seperti manusia. Kecerdasan buatan tidak terbatas pada objek fisik semata, seperti robot. Sebaliknya, kecerdasan buatan mengacu pada program-program yang memiliki basis matematis atau instruksional, berbeda dengan program-program konvensional yang bertindak berdasarkan instruksi. Saat ini, AI telah merambah ke berbagai bidang, mulai dari yang umum seperti pembelajaran dan persepsi, hingga bidang-bidang khusus seperti deteksi objek manusia, diagnosis penyakit dan lainnya (Rangkuti, 2023).

Salah satu implementasi dari kecerdasan buatan (*Artificial Intelligence*) saat ini yaitu penggunaan AI sebagai pendeteksi objek atau manusia untuk meningkatkan keamanan. Keamanan merupakan aspek kritis dalam menjaga suatu area atau fasilitas, terutama di tempat-tempat yang memiliki kebijakan akses terbatas. Area seperti perusahaan, gedung-gedung pemerintahan, atau fasilitas-fasilitas khusus memerlukan kontrol akses yang ketat untuk mencegah masuknya orang yang tidak berizin. Sistem kontrol akses manual dengan petugas keamanan sering kali kurang efisien dan rentan terhadap kesalahan manusia atau pelanggaran keamanan. Dengan adanya teknologi sistem deteksi objek manusia, proses ini dapat ditingkatkan secara signifikan.

Dengan semakin berkembangnya teknologi, sistem deteksi objek manusia telah menjadi area penelitian yang penting dalam berbagai aplikasi seperti keamanan, dan pemantauan. Berkembangnya teknologi kamera seperti adanya kamera *depth sensor* yang tidak terlalu terpengaruh kondisi pencahayaan dibandingkan kamera biasa, juga menunjang sistem deteksi yang lebih akurat. CV. Ateri Global Teknologi merupakan sebuah perusahaan yang bergerak dibidang

teknologi. Mereka menyediakan layanan pengembangan perangkat keras serta perangkat lunak untuk klien mereka. Untuk memfasilitasi pelayanan tersebut, kantor dari CV. Ateri Global Teknologi menyimpan berbagai peralatan seperti komputer, laptop, dan sebagainya. Keamanan di CV. Ateri Global Teknologi sangat penting mengingat risiko kejahatan yang mungkin terjadi seperti pencurian peralatan tersebut, terutama pada malam hari ketika tidak ada pegawai yang berada di kantor. Hal ini dikarenakan tidak adanya sistem ataupun personil keamanan khusus yang menjaga kantor pada malam hari. Dengan memperketat pengawasan pada jam-jam tertentu, CV. Ateri Global Teknologi dapat mengurangi potensi terjadinya insiden tersebut. Melalui diskusi yang telah dilakukan dengan pihak CV. Ateri Global Teknologi, diputuskan untuk mengembangkan sistem deteksi objek manusia sebagai sistem keamanan untuk memenuhi kebutuhan tersebut.

Dengan memahami latar belakang tersebut, maka penulis melakukan penelitian yang berjudul "SISTEM DETEKSI OBJEK MANUSIA MENGGUNAKAN ALGORITMA YOLO DAN KAMERA DEPTH SENSOR (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)".

1.1 Rumusan Masalah

Berdasarkan latar belakang yang ada, permasalahan yang dapat di angkat adalah:

1. Bagaimana mengembangkan sistem deteksi objek manusia di CV. Ateri Global Teknologi?
2. Bagaimana mengimplementasikan metode yang dapat mendeteksi objek manusia pada kamera *depth sensor*?

1.2 Tujuan Penelitian

Mengacu pada rumusan masalah yang diambil, adapun tujuan dalam penelitian ini adalah:

1. Mengembangkan sistem pendeteksian objek manusia di CV. Ateri Global Teknologi dengan memanfaatkan kamera *depth sensor*.
2. Mengimplementasikan algoritma YOLOv8 untuk mendeteksi objek deteksi manusia.

1.3 Batasan Masalah

Ruang lingkup pembahasan dalam penelitian ini adalah:

1. Metode yang digunakan dalam penelitian ini adalah Algoritma YOLOv8.
2. Penelitian ini membatasi diri pada kamera *depth sensor* sebagai sumber data utama, tanpa memasukkan data dari jenis *frame* lainnya.
3. Penelitian ini membatasi skenario pengujian pada kondisi tertentu, seperti kondisi pencahayaan dan suhu tertentu, untuk fokus pada aspek-aspek tertentu dari deteksi objek manusia.

1.4 Manfaat Penelitian

Penelitian sistem deteksi objek manusia menggunakan algoritma YOLOv8 dan kamera *depth sensor* memiliki sejumlah manfaat yang dapat diidentifikasi, antara lain:

1. Manfaat Bagi Mahasiswa
   a. Memberikan dukungan kepada mahasiswa dalam menerapkan pengetahuan yang telah diperoleh selama studi di perguruan tinggi, sehingga dapat meningkatkan kesiapan mereka untuk memasuki lingkungan kerja.
   b. Memperluas pemahaman tentang pengembangan metode sistem deteksi objek manusia yang efisien dan akurat dengan menggunakan Algoritma YOLOv8 dan kamera *depth sensor*.
2. Bagi Program Studi Teknik Informatika
   a. Dapat digunakan sebagai referensi untuk perpustakaan dan sebagai sumber pengetahuan tambahan bagi pembaca.
3. Bagi Tempat Penelitian
   a. Penelitian ini dapat memberikan kontribusi signifikan terhadap peningkatan keamanan pada area-area yang memiliki kebijakan akses terbatas.
   b. Model yang dihasilkan dari penelitian ini dapat diterapkan dalam berbagai lingkungan, terutama di tempat-tempat dengan kondisi pencahayaan yang rendah atau berubah-ubah, di mana metode deteksi konvensional mungkin kurang efektif.

1.5 Tempat Penelitian

Tempat penelitian ini dilakukan yaitu di :

Tempat : CV. Ateri Global Teknologi

Alamat. : Jalan Ibu Sangki, Graha Alamanda Estate Blok D4, Kecamatan Cibeber Kelurahan Cimahi Selatan Kota Cimahi, 40531

Telepon : +62 855-7190-021

1.6 Metodologi Penelitian

Metodologi penelitian yang akan dilakukan dalam penelitian ini adalah:

1. Studi Literatur

   Tahapan ini melakukan pencarian dan memahami dengan mengumpulkan referensi yang dibutuhkan untuk memperoleh informasi dan data yang berkaitan dengan Algoritma YOLOv8, dan teori dasar mengenai kamera *depth sensor*. Referensi yang dapat di ambil dapat berupa artikel, buku maupun jurnal penelitian.

2. Perancangan Sistem

   Tahap ini melakukan perancangan sistem dengan algoritma YOLOv8.

3. Simulasi

   Tahapan ini melakukan simulasi algoritma YOLOv8 juga melakukan validasi tingkat akurasi sistem.

4. Analisis dan Evaluasi

   Tahapan ini dilakukan analisis tingkat performansi dan akurasi sistem deteksi manusia menggunakan algoritma YOLOv8 dan kamera *depth sensor*.

1.7 Sistematika Penulisan

Untuk memudahkan memadukan argumentasi dan bukti pendukung dalam penelitian ini, ada beberapa bab yang disusun antara lain, yaitu:

BAB I PENDAHULUAN

Bab ini membahas latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, metode penelitian, hingga sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas konsep dasar dari sistem deteksi, manusia, algoritma YOLOv8 dan kamera depth sensor.

BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan membahas tentang kebutuhan sistem yang akan digunakan dalam penerapan perancangan sistem dalam pengembangan sistem deteksi menggunakan algoritma YOLOv8 dan kamera depth sensor.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menjelaskan tentang implementasi sistem deteksi objek manusia yang dirancang dengan menggunakan algoritma YOLOv8 dan kamera depth sensor berupa langkah pengujian sistem, hasil dan evaluasi sistem.

BAB V KESIMPULAN DAN SARAN

Pada bagian ini dijelaskan kesimpulan dan saran untuk pengembangan selanjutnya.

BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Terdapat beberapa penelitian terdahulu yang membahas tentang penggunaan algoritma YOLO dan kamera depth sensor dalam sistem deteksi objek manusia, namun tidak ditemukan penelitian yang secara spesifik menggunakan algoritma YOLOv8 dan kamera depth sensor. Beberapa penelitian terdahulu yang relevan antara lain:

Penelitian (RAHMAN, et al., 2022) ini membuat sistem deteksi objek manusia menggunakan algoritma YOLOv4 dengan kamera termal. Dataset yang digunakan sebanyak 2535 frame dengan beberapa jarak ukur mulai dari 5m, 10m, 15m hingga 20m. Untuk mendapatkan nilai mAP (Mean Average Precision) sistem yang maksimal menggunakan pre- processing pada frame berupa resizing, mengubah rasio dengan membagi data latih dan data uji serta mengubah nilai hyperparameter seperti learning rate dan batch size. Selain mAP (Mean Average Precision), beberapa parameter kinerja yang diuji adalah precision, recall, dan f1-score. Hasil akhir tugas akhir ini mampu membuat sistem yang bisa mendeteksi objek khususnya manusia berbasis frame termal pada malam hari dengan nilai mAP (Mean Average Precision) yang dihasilkan mencapai 86.51% dengan konfigurasi sistem yang digunakan adalah perbandingan rasio data latih dan data uji 90% banding 10%, resize frame menjadi 512 × 512 piksel, learning rate 0.01, dan batch size 32. Kata kunci: Pencurian, kamera termal, You Only Look Once version 4 (YOLOv4).

Penelitian (Yanto, et al., 2023) ini berjudul “Yolo-V8 Peningkatan Algoritma Untuk Deteksi Pemakaian Masker Wajah”, bertujuan untuk memfasilitasi deteksi masker dan memastikan bahwa masker digunakan dengan benar, sehingga memastikan keselamatan dan kesehatan semua orang di lingkungan dengan pendekatan AI menggunakan algoritma YOLOv8. Hasilnya menunjukkan tingkat akurasi yang tinggi yaitu 94% untuk kelas badmask, 97% untuk mask, dan 95% untuk kelas nomask. Nilai F1-Confidence, Precision, dan Recall untuk semua

kelas juga tinggi, masing- masing sebesar 0,94, 0,96, dan 0,978. Waktu komputasi rata-rata yang dibutuhkan oleh algoritma hanya 17ms. masing-masing sebesar 0,94, 0,96, dan 0,978. Waktu komputasi rata-rata yang dibutuhkan oleh algoritma hanya 17ms. masing-masing sebesar 0,94, 0,96, dan 0,978. Waktu komputasi rata-rata yang dibutuhkan oleh algoritma hanya 17ms.

Penelitian (Bai, et al., 2023) yang berjudul Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery, Studi ini menyajikan analisis komprehensif dan peningkatan algoritma YOLOv8-n untuk deteksi objek, dengan fokus pada integrasi strategi Wasserstein Distance Loss, FasterNext, dan Context Aggravation. Melalui studi ablasi yang mendetail, setiap strategi dievaluasi secara sistematis secara individual dan kolektif untuk menilai kontribusinya terhadap kinerja model. Hasilnya menunjukkan bahwa setiap strategi secara unik meningkatkan kinerja model, meningkatkan peta secara signifikan, dan mengurangi kompleksitas model ketika ketiganya diintegrasikan. Visualisasi melalui Grad-CAM semakin memperkuat kapasitas model yang ditingkatkan untuk mengekstrak dan fokus pada fitur objek utama. Dibandingkan dengan model yang ada, seperti YOLOv5-n, YOLOv5-s, YOLOX-n, YOLOX-s, dan YOLOv7-tiny, model YOLOv8-n yang ditingkatkan mencapai keseimbangan optimal antara akurasi dan kompleksitas model, mengungguli model lain dalam hal akurasi model, kompleksitas model, dan kecepatan inferensi model. Pengujian inferensi gambar lebih lanjut memvalidasi performa model, menunjukkan kemampuan deteksi yang unggul.

Penelitian (Drantantiyas, et al., 2023) yang berjudul Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8, Tujuan penelitian ini adalah menentukan performansi model sistem penghitung jumlah kepala dengan menggunakan algoritma Yolov8. Penelitian ini hanya berfokus membuat model deteksi jumlah orang. Jumlah dataset yang dirancang berjumlah 2390 gambar yang diperoleh dari dataset Roboflow, dengan pemisahan data sebesar 70:20:10 untuk masing-masing, data latih; data uji; data validasi. Besar Epoch pada pelatihan model yang digunakan adalah 50. Algoritma deteksi jumlah kepala meggunakan YOLOv8. Nilai yang diukur adalah performasi dari model data training, nilai confusion matrix dan nilai

evaluasi dari confusion matrix. Nilai evaluasi yang akan dihitung adalah nilai presisi, nilai akurasi, recall dan F1-score. Diperoleh hasil pengujian nilai akurasi sebesar 87,56 %, nilai presisi 83,74%, nilai recall 100% dan nilai F1-score 91,15%. Kurva presisi memberikan nilai tertinggi 1 pada tingkat kepercayaan 0,857, recall bernilai 0,8 pada tingkat kepercayaan 0, F1 0,716 pada kepercayaan 0,36 dan presisi-recall 0,771 pada 0,5 mAP. Berdasarkan nilai ini, model sudah cukup mendeteksi jumlah kepala.

Penelitian (Zhou, et al., 2022) yang berjudul Human Position Detection Based on Depth Camera Image Information in Mechanical Safety, Perangkat yang digunakan untuk mendeteksi posisi manusia dalam keselamatan mekanis terutama mencakup tirai lampu pengaman, pemindai laser pengaman, bantalan pengaman, dan sistem penglihatan. Namun, perangkat ini mungkin dilewati saat digunakan, dan manusia atau peralatan tidak dapat dibedakan. Untuk mengatasi masalah ini, kamera kedalaman diusulkan sebagai alat pendeteksi posisi manusia dalam keselamatan mekanis. Proses pendeteksian posisi manusia berdasarkan informasi gambar kamera kedalaman diberikan; ini terutama mencakup perolehan informasi gambar, deteksi keberadaan manusia, dan pengukuran jarak. Sementara itu, metode deteksi posisi manusia berdasarkan kamera kedalaman Intel RealSense dan algoritma MobileNet-SSD diusulkan dan diterapkan untuk perlindungan keselamatan robot. Hasilnya menunjukkan bahwa informasi gambar yang dikumpulkan oleh kamera kedalaman dapat mendeteksi posisi manusia secara real time yang dapat menggantikannya perangkat pendeteksi posisi manusia keselamatan mekanis yang ada. Pada saat yang sama, kamera kedalaman hanya dapat mendeteksi objek manusia tetapi tidak dapat mendeteksi perangkat seluler dan mewujudkan pemisahan serta peringatan dini terhadap manusia dan perangkat seluler.

Penelitian (Motwani & S, 2023) yang berjudul Human Activities Detection using DeepLearning Technique- YOLOv8, Model YOLOv8 untuk mengonfirmasi penerapannya secara keseluruhan, pada dua kumpulan data yaitu FDDB & MASKER. Ini membantu untuk memeriksa perilaku fitur dari kumpulan data Masker, yang ditujukan untuk Masker COVID-19 Deteksi saja. Mask adalah

dataset utama dalam percobaan ini. Di atas ini, kumpulan data ImageNet digunakan untuk pra-pelatihan dan FDDB (Face Detection Dataset & Benchmarks) dataset untuk mengenali wajah a manusia. Ketepatan model pada FDDB adalah 58,9% & seterusnya Kumpulan data MASK adalah 66,5%.

2.2 Pengertian Sistem

Sistem adalah himpunan atau suatu kumpulan dari unsur, komponen, atau variabel yang terorganisasi, saling tergantung satu sama lain, saling berinteraksi, serta terpadu. Secara umum, sistem bisa dipahami menjadi sebuah kesatuan yang dibangun dari beberapa komponen, dan elemen penggerak untuk mencapai maksud tertentu. Elemen pada sistem saling bekerjasama dan berhubungan guna mencapai tujuan yang diharapkan. Menurut Andri Kristanto (Kristanto, 2008) Sistem merupakan bentuk dari unsur-unsur tertentu, pada setiap sistem terdiri dari empat unsur di dalamnya, yakni:

1. Komponen/Komponen Utama: Bagian-bagian yang membentuk sistem dan memiliki peran penting dalam menjalankan fungsi sistem. Contohnya dalam sistem komputer, komponen utama dapat mencakup CPU, RAM, hard drive, dan sebagainya.
2. Tujuan: Tujuan atau hasil yang ingin dicapai oleh sistem. Tujuan ini biasanya menjadi alasan utama mengapa sistem dibangun atau diperlukan.
3. Input: Data atau masukan yang diberikan kepada sistem. Ini dapat berupa informasi, energi, atau bahan mentah yang diperlukan oleh sistem untuk melakukan operasi.
4. Proses/Operasi: Proses atau aktivitas yang dilakukan oleh sistem untuk mengolah input menjadi output atau mencapai tujuan. Ini bisa berupa serangkaian tindakan atau algoritma yang dijalankan oleh komponen sistem.
5. Output: Hasil dari operasi sistem. Output ini dapat berupa informasi yang dihasilkan, produk yang diproduksi, atau tindakan yang diambil sebagai respons terhadap input.

6. Umpan Balik *(Feedback)*: Informasi yang diberikan kepada sistem tentang kinerjanya. Ini membantu sistem untuk memantau dan mengatur dirinya sendiri agar mencapai tujuan yang diinginkan.
7. Lingkungan: Sumber daya eksternal dan faktor-faktor yang ada di sekitar sistem dan dapat memengaruhi atau dipengaruhi oleh sistem. Lingkungan dapat bersifat fisik, sosial, ekonomi, atau teknologis.
8. Batas *(Boundary)*: Batas yang memisahkan sistem dari lingkungannya. Ini menentukan apa yang termasuk dalam sistem dan apa yang dianggap sebagai lingkungan eksternal.
9. Interkoneksi/Interaksi: Hubungan dan koneksi antara komponen-komponen dalam sistem. Interaksi ini memungkinkan komponen-komponen untuk bekerja sama dalam mencapai tujuan sistem.
10. Waktu: Faktor waktu juga dapat menjadi unsur penting dalam sistem, terutama dalam konteks perencanaan, pengukuran, dan manajemen kinerja.

Pada unsur-unsur ini bekerjasama untuk membentuk sistem yang berfungsi, dan pemahaman yang baik tentang unsur-unsur ini penting dalam analisis, perancangan, dan pengelolaan sistem apa pun.

2.3 Deteksi

Deteksi adalah proses mengidentifikasi atau menemukan keberadaan atau sifat suatu objek, peristiwa, atau keadaan tertentu dalam suatu lingkungan atau konteks tertentu. Dalam berbagai konteks, deteksi dapat merujuk pada berbagai jenis objek atau peristiwa yang ditemukan atau diidentifikasi. Penerapan deteksi sering melibatkan penggunaan sensor, teknologi pengindraan, atau sistem komputasi untuk menganalisis data dan memberikan peringatan atau informasi tentang keberadaan atau sifat suatu objek atau peristiwa (Sutabri, 2012).

Tujuan deteksi adalah menyelesaikan suatu permasalahan dengan berbagai metode yang diterapkan, tergantung pada pendekatan yang digunakan, untuk menghasilkan solusi (Sutabri, 2012).

2.4 Manusia

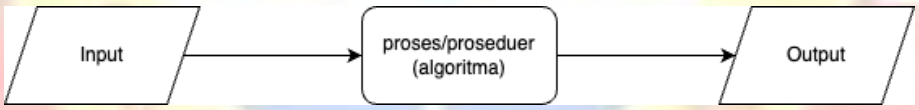
Definisi manusia menurut Kamus Besar Bahasa Indonesia (KBBI) adalah makhluk yang memiliki akal budi dan mampu menguasai makhluk lain. Manusia melalui lima tahap kehidupan yang mencakup masa bayi, anak, remaja, dewasa, hingga lanjut usia (lansia). Lansia diartikan sebagai individu yang berusia 60 tahun ke atas menurut definisi Organisasi Kesehatan Dunia (WHO) tahun 2014. Gambar 2.1 merupakan ilustrasi dari tahap kehidupan manusia.

Gambar 2.1 Ilustrasi Tahap Kehidupan Manusia

2.5 Algoritma

Algoritma dapat dijelaskan sebagai rangkaian operasi yang dipikirkan secara logis dan teratur untuk menyelesaikan suatu masalah dan menghasilkan hasil yang diinginkan. Dalam konteks komputasi, algoritma sangat vital karena berfungsi sebagai panduan sistematis yang diperlukan oleh komputer. Algoritma yang efektif mirip dengan menggunakan peralatan yang tepat di bengkel, sementara algoritma yang tidak efektif sama seperti mencoba memotong kayu dengan gunting, yang jelas tidak akan efisien dan memakan waktu (Kani, 2020).

Algoritma merupakan metode pemecahan masalah yang terbatas, deterministik, dan efektif dalam ilmu komputer atau informatika. Ini merupakan bagian penting dari ilmu komputer dan menjadi fokus utama penelitian. Algoritma menggambarkan langkah-langkah pemecahan masalah dengan bahasa yang sangat alami, kemudian diimplementasikan ke dalam program komputer untuk meningkatkan efisiensi penyelesaian masalah. Pemilihan bahasa pemrograman untuk mengimplementasikan algoritma sangat tergantung pada preferensi dan keahlian individu programmer.

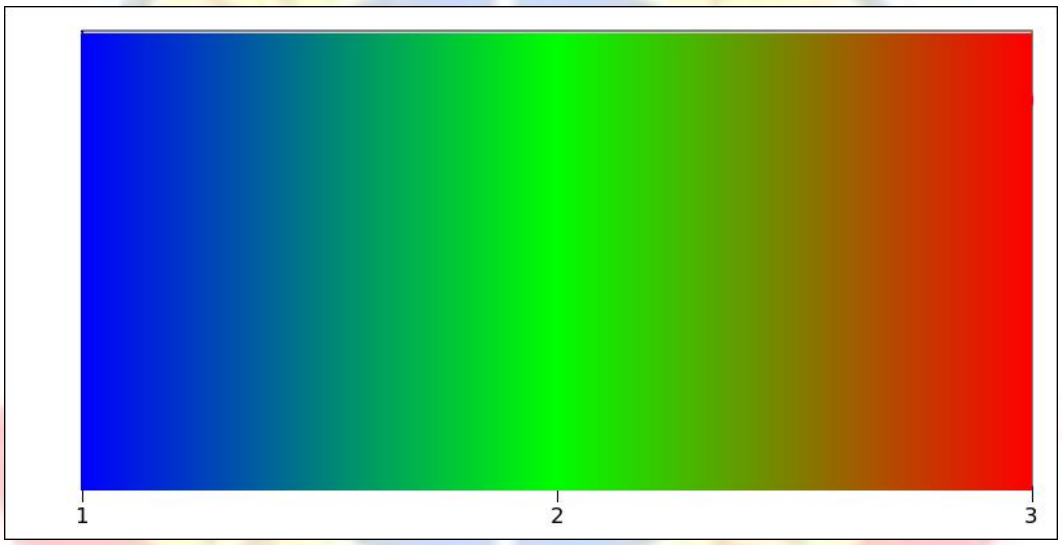


Gambar 2.2 Proses Pemecahan Masalah (Algoritma)

Pada Gambar 2.2 diilustrasikan proses dalam pemecahan masalah atau algoritma. Tahapan dari proses tersebut terdiri dari tiga tahap yaitu input, kemudian input tersebut diproses, lalu proses tersebut menghasilkan output.

2.6 RGB

RGB adalah suatu model warna yang terdiri atas 3 buah warna: merah (Red), hijau (Green), dan biru (Blue), yang ditambahkan dengan berbagai cara untuk menghasilkan bermacam-macam warna seperti pada Gambar 2.3 Warna RGB. Kegunaan utama model warna RGB adalah untuk menampilkan citra / gambar dalam perangkat elektronik, seperti televisi dan komputer, walaupun juga telah digunakan dalam fotografi biasa. Sebelum era elektronik, model warna RGB telah memiliki landasan yang kuat berdasarkan pemahaman manusia terhadap teori trikromatik.



Gambar 2.3 Warna RGB

2.7 Deep Learning

Deep learning merupakan suatu bidang dari machine learning, pengembangan Multilayer Neural Network untuk memungkinkan deteksi objek, pengenalan suara, dan terjemahan bahasa. Dikembangkan pada tahun 1950, kemudian dapat di aplikasikan dengan sukses pada tahun 1990 (Putro, et al., 2000). Metode deep learning terdiri dari berbagai lapisan atau dalam mempelajari karakteristik data dengan berbagai tingkat abstraksi (Openg, et al., 2022). Deep learning memberikan arsitektur yang sangat baik untuk supervised learning. Oleh karena itu, deep learning dianggap sebagai bagian dari machine learning, deep learning terdiri dari banyak tingkatan proses informasi non-linear dan linear untuk memungkinkan supervised learning, unsupervised learning dan semi supervised learning (Santoso & Ariyanto, 2018). Salah satu jenis algoritma jaringan saraf tiruan yang menggunakan data adalah deep learning. Algoritma ini menerima data sebagai masukan dan kemudian memprosesnya dengan menggunakan sejumlah hidden layer untuk melakukan transformasi non linier pada data masukan, kemudian algoritma ini untuk menghasilkan nilai keluaran (N.Nufus, et al., 2021).

Perbedaan Supervised learning dan Unsupervised learning (RevoU, 2024) terdapat pada Tabel 2.1 Perbedaan Supervised dan Unsupervised learning.

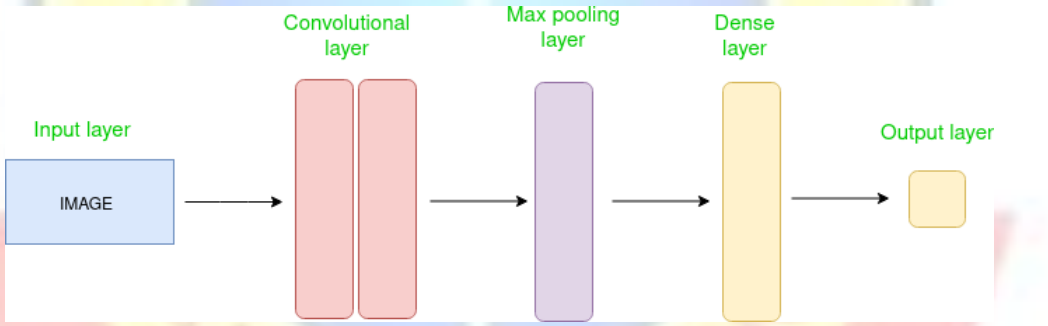
Tabel 2.1 Perbedaan Supervised dan Unsupervised learning

	Supervised Learning	Unsupervised Learning
Input Data	Data berlabel	Data tidak berlabel
Kompleksitas Komputasi	Lebih sederhana	Lebih kompleks
Keakuratan	Sangat Akurat	Kurang Akurat
Jumlah Kelas	Jumlah kelas diketahui	Jumlah kelas tidak diketahui

2.8 CNN (Convolutional Neural Network)

Convolutional Neural Network merupakan salah satu algoritma deep learning yang dikembangkan dari Multilayer Percepton (MLP), dirancang untuk mengolah data dua dimensi. Algoritma ini digunakan untuk klasifikasi data yang sudah diberi label menggunakan metode supervised learning dengan cara kerjanya yang memerlukan data yang dilatih dan memiliki variable yang sudah ditargetkan, tujuan dari metode ini adalah pengelompokan suatu data menuju data yang sudah ada. (Ilahiyah & Nilogiri, 2018)

Jaringan konvolusi adalah jaringan khusus yang dimiliki CNN, citra masukan diolah pada lapisan ini berdasarkan filter yang sudah ditentukan. Hasil dari setiap lapisan ini berupa sebuah pola dari beberapa bagian citra yang nanti akan diklasifikasikan.



Gambar 2.4 Arsitektur CNN

Pada Gambar 2.4 terdapat arsitektur dari CNN yang terdiri dari beberapa layer (geeksforgeeks, 2024), yaitu:

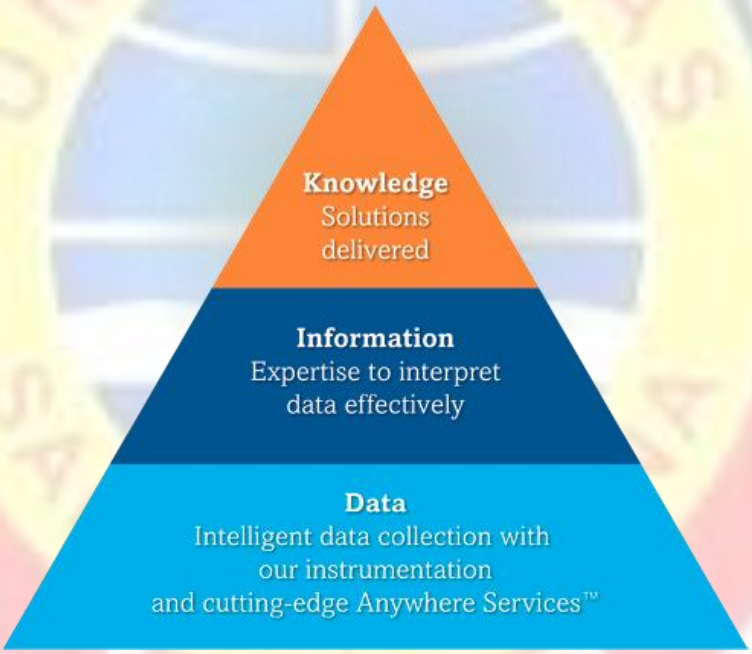
1. Input layer, menerima input berupa gambar yang direpresentasikan sebagai nilai piksel. Dimensi data input ditentukan oleh tinggi, lebar, dan format warna (contohnya RGB).
2. Convolutional layer, melakukan ekstraksi fitur dengan menerapkan operasi konvolusi pada data input. Operasi ini terdiri dari beberapa filter (dikenal

juga sebagai kernel) yang meluncur di atas data input, menghasilkan peta fitur yang menyoroti pola penting seperti tepi, tekstur, atau bentuk.

3. Max pooling layer. Layer ini mengurangi dimensi spasial dari peta fitur sambil mempertahankan informasi yang paling penting. Ini bekerja dengan membagi input menjadi wilayah yang tidak tumpang tindih dan mempertahankan hanya nilai maksimum dalam setiap wilayah, secara efektif melakukan pengurangan sampel dari peta fitur.
4. Dense layer. Setelah layer konvolusi dan pooling, satu atau lebih layer dense ditambahkan untuk melakukan tugas klasifikasi atau regresi. Setiap neuron dalam layer dense terhubung ke setiap neuron dalam layer sebelumnya, memungkinkan jaringan untuk belajar fitur tingkat tinggi dan membuat prediksi.
5. Output Layer. Setelah layer konvolusi dan pooling, satu atau lebih layer dense ditambahkan untuk melakukan tugas klasifikasi atau regresi. Setiap neuron dalam layer dense terhubung ke setiap neuron dalam layer sebelumnya, memungkinkan jaringan untuk belajar fitur tingkat tinggi dan membuat prediksi.

2.9 Data, Informasi, Pengetahuan

Data adalah bilangan, terkait dengan angka angka atau atribut atribut yang bersifat kuantitas, yang berasal dari hasil observasi, eksperimen, atau kalkulasi. Informasi adalah data di dalam satu konteks tertentu. Informasi merupakan kumpulan data dan terkait dengan penjelasan, interpretasi, dan berhubungan dengan materi lainnya mengenai objek, peristiwa peristiwa atau proses tertentu. Sementara itu, pengetahuan adalah informasi yang telah diorganisasi, disintesisksn, diringkaskan untuk meningkatkan pengertian, kesadaran atau pemahaman (Bergeron, 2003).



Gambar 2. 5 Ilustrasi Data, Informasi dan Pengetahuan

2.10 Data Preprocessing

Data yang belum diproses disebut data mentah, dimana data harus disiapkan terlebih dahulu sebelum dapat dipakai dalam suatu proses. Data mentah atau data real cenderung mengandung kesalahan atau mengandung nilai-nilai yang menyimpang dari yang diharapkan (Kumar & Chadha, 2012). Data yang mengandung kesalahan dikarenakan data tersebut tidak lengkap ataupun tidak konsisten. Ketidaklengkapan data terjadi karena adanya atribut data yang tidak tersedia, hilangnya nilai untuk beberapa data (atribut) karena adanya penghapusan data yang dianggap tidak penting. Sedangkan data dianggap tidak konsisten karena pada saat pengumpulan data adanya instrumen yang rusak karena kesalahan manusia(human error) ataupun kesalahan komputer, adanya ketidaksamaan (tidak konsisten) dalam penamaan suatu data dengan data yang lain, yang merupakan suatu data yang sama.

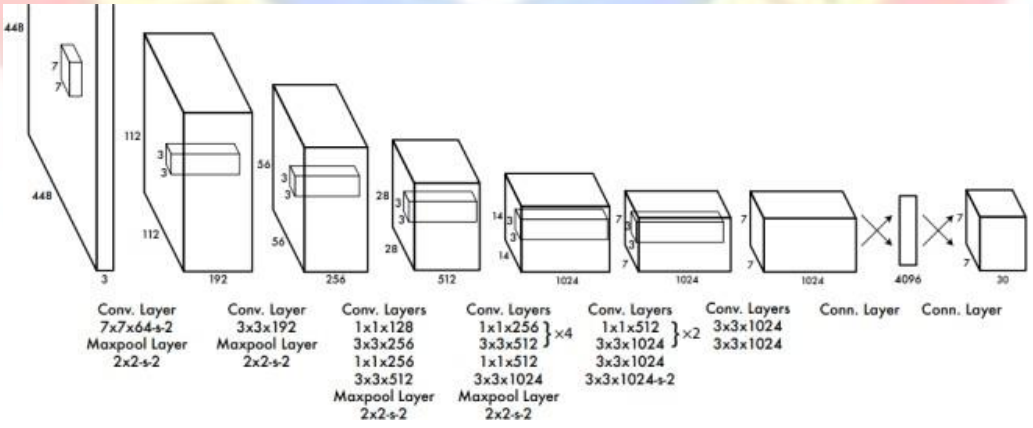
Preprocessing data merupakan langkah penting dalam proses penemuan pengetahuan, karena keputusan-keputusan yang berkualitas harus didasarkan pada data yang berkualitas (Kumar & Chadha, 2012). Preprocessing data sering kali digunakan untuk mengurangi kesalahan data dan sistematis bias dalam data mentah sebelum analisis apapun terjadi.

2.11 Algoritma YOLO (You Only Look Once)

*You Only Look Onc*e (YOLO) adalah salah satu pendekatan untuk melakukan pendeteksian objek secara real-time berbasis Convolutional Neural Network. YOLO menggunakan pendekatan jaringan syaraf tunggal (Single neural network) untuk melakukan pendeteksian objek pada sebuah frame. Jaringan ini menggunakan fitur dari semua gambar untuk memprediksi setiap bounding box yang dapat melakukan prediksi pada kotak-kotak pembatas dan probabilitas secara langsung dalam satu evaluasi (Redmon, et al., 2015).

Algoritma YOLO memiliki beberapa versi mulai dari YOLO, YOLOv2, YOLOv3, scaled-YOLOv4 hingga YOLOv8. Algoritma YOLO hingga YOLOv4 sudah banyak digunakan untuk penelitian deteksi objek, maka pada penelitian ini menggunakan algoritma YOLOv8 dapat meng-identifikasi objek secara cepat, efisien, dan akurat terbukti dengan nilai AP yang mencapai angka 65%.

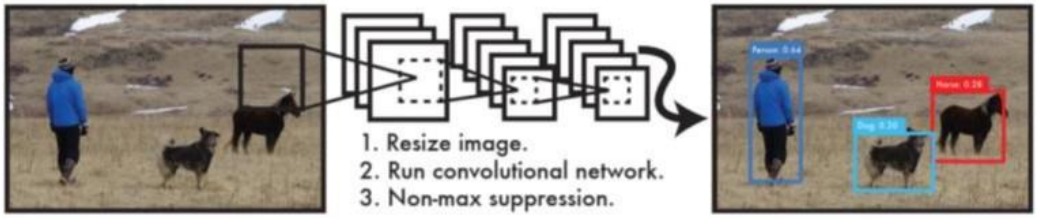
Jaringan deteksi YOLO memiliki 24 lapisan konvolusi (convolutional layer) yang diikuti oleh 2 lapisan yang terhubung penuh (fully connected layer). Beberapa lapisan konvolusi menggunakan lapisan reduksi 1x1 sebagai alternative dalam mengurangi kedalaman feature maps yang diikuti oleh 3x3 lapisan konvolusional (convolutional layer) seperti pada Gambar 2.6.



Gambar 2.6 Arsitektur YOLO

Terdapat tiga langkah dalam mendeteksi objek menggunakan YOLO yang diilustrasikan pada Gambar 2.7.

1. Mengubah ukuran masukan gambar menjadi 448x448
2. Menjalankan jaringan syaraf tunggal (Single neural network) pada gambar
3. Melakukan threshold pada hasil deteksi berdasarkan nilai confidence



Gambar 2.7 Sistem deteksi YOLO

YOLO membagi gambar input menjadi grid S x S. Jika pusat suatu objek jatuh ke dalam sel grid, maka sel grid bertanggung jawab untuk mendeteksi objek tersebut. Setiap sel grid memprediksi kotak pembatas (bounding boxes) B dan nilai keyakinan (confidence score) untuk kotak tersebut, serta probabilitas kelas kondisional C. Nilai keyakinan (confidence score) ini menggambarkan seberapa akurat kotak tersebut menurut perkiraannya. YOLO mendefinisikan confidence sebagai Pr(objek).

Jika tidak ada objek yang terdeteksi pada sel, nilai keyakinan akan bernilai nol. Jika tidak, sistem ingin nilai keyakinan sama dengan Intersection Over Union (IoU) antara kotak prediksi dan ground truth. $IoU\ pred\ truth$

Setiap kotak pembatas B terdiri dari 5 komponen x, y, w, h, seperti pada Gambar 2.8, dan confidence. Nilai koordinat (x, y) menyatakan pusat kotak, relative terhadap batas kotak grid. Nilai koordinat kemudian dinormalisasi untuk jatuh di antara 0 dan 1. Lebar (w) dan tinggi (h) relative terhadap keseluruhan gambar, dan dinormalisasikan juga. Nilai keyakinan (confidence score) menyatakan seberapa yakin model tersebut, bahwa kotak pembatas B berisi sebuah objek dan seberapa akurat menurutnya kotak yang ia prediksi. Oleh karena itu, prediksi YOLO memiliki keluaran vector S, S, B, 5+C.

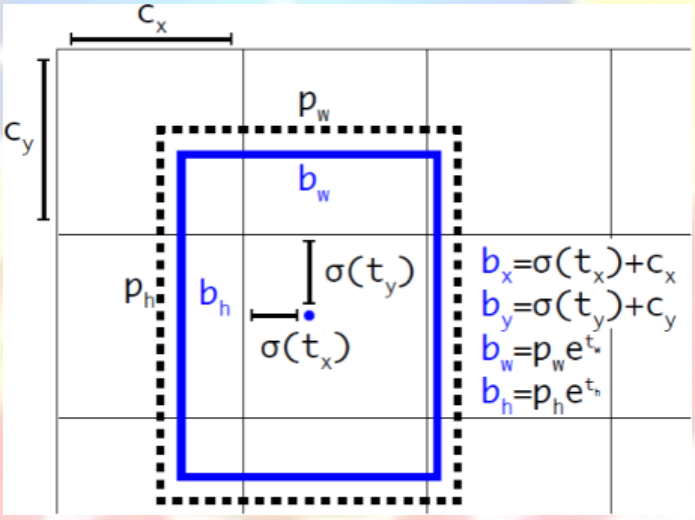
Faktor penentuan untuk mendapatkan prediksi akhir adalah class confidence score, berdasarkan probabilitas kondisional kelas dan box confidence score. Class confidence score mengukur nilai kepercayaan terhadap klasifikasi dan lokalisasi objek. Class confidence score memberi nilai kepercayaan kelas spesifik untuk setiap kotak, yang mengkodekan kemungkinan kelas yang muncul di kotak dan seberapa sesuainya kotak yang diprediksi dengan objek. Persamaan pada class confidence score untuk setiap kotak prediksi ditunjukkan pada Persamaan (2.1).

$$\Pr(Class\ i\ |Object)\ .\ \Pr Object\ .\ IoU\ pred\ truth = \Pr(Class\ i\ ).\ IoU\ pred\ truth \quad (2.1)$$

Keterangan:

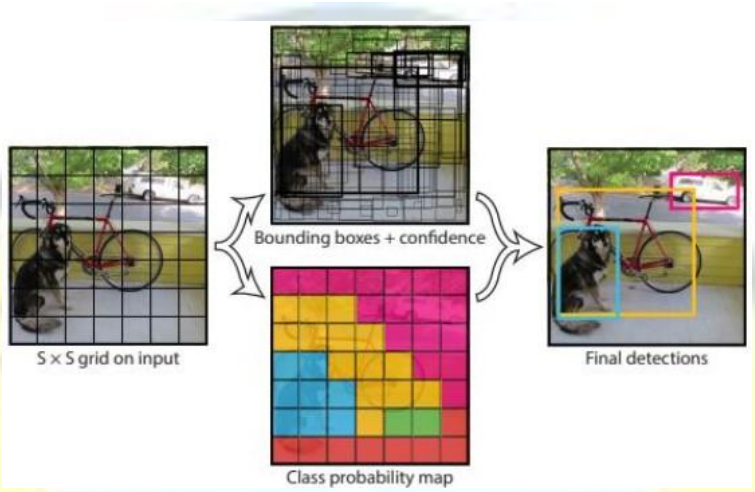
$\Pr(Class\ i\ |Object)$: probabilitas kondisional kelas i.

$\Pr Object$: probabilitas kelas i.



Gambar 2.8 Bounding Box YOLO

Dari persamaan tersebut, akan mendapatkan nilai confidence dari kelas spesifik. Nilai ini akan merepresentasikan probabilitas kelas yang muncul didalam kotak dan seberapa akurat kotak yang diprediksi. Seperti pada Gambar 2.8 YOLO mendeteksi model sebagai regresi. Hal ini membagi gambar menjadi grid dan setiap grid memprediksi bounding boxes, nilai confidence dari setiap kotak dan kelas probabilitas.

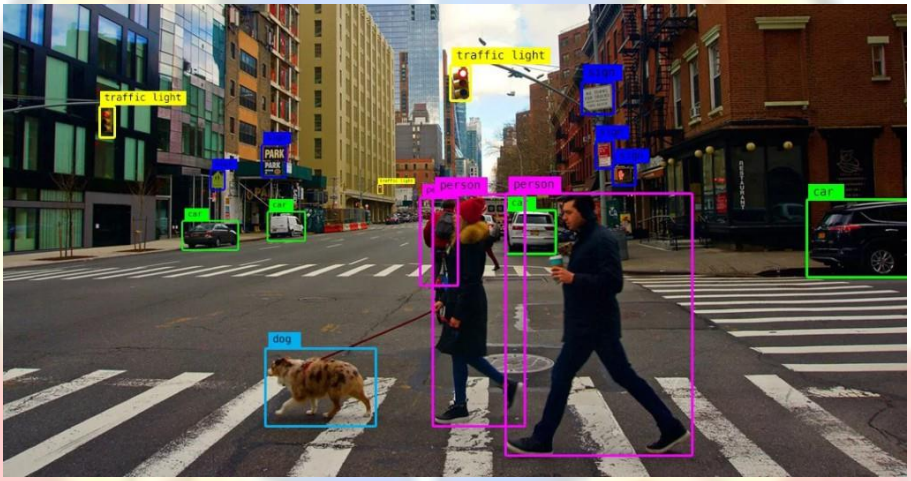


Gambar 2.9 Proses Deteksi pada YOLO

Penelitian ini menggunakan YOLOv3 dimana Darknet53 sebagai feature extractor. Darknet53 merupakan jaringan baru untuk melakukan ekstraksi fitur. Dengan menggunakan 53 layer seperti pada Gambar 2.9 berbeda dengan versi sebelumnya YOLOv2 yang menggunakan Darknet-19. Pada jaringan Darknet53 menggunakan berturut-turut 3x3 dan 1x1 lapisan konvolusi.

2.12 Computer Vision

Computer vision adalah bidang ilmu komputer yang bertujuan untuk memberikan kemampuan pada komputer atau mesin untuk memahami, menganalisis, dan menafsirkan informasi visual dari gambar dan video. Tujuan utama dari computer vision adalah untuk mengembangkan sistem yang dapat melihat dan memahami dunia sekitarnya dengan cara yang mirip dengan cara manusia melakukannya. Hal Ini melibatkan identifikasi objek seperti pada Gambar 2.10, deteksi dan pelacakan gerakan seperti pada Gambar 2.11, pengenalan wajah, rekonstruksi 3D, analisis citra medis dan pengenalan tulisan tangan (Suradi, et al., 2023).



Gambar 2.10 Identifikasi Objek

Gambar 2.11 Pelacakan Gerakan

2.13 OpenCV

OpenCV merupakan salah satu library yang memiliki fungsi-fungsi pemrograman pada computer vision secara real-time. OpenCV bersifat open-source yang dapat digunakan untuk membantu hal-hal yang bersifat akademis dan komersil. Pada OpenCV terdapat antarmuka untuk C, C++, Python, dan Java yang dapat dijalankan pada Windows, Mac, Linux, dan Android OpenCV memiliki lebih dari 2500 algoritma yang telah dioptimalkan (Budiarjo, 2020).

OpenCV memiliki banyak sekali fitur yang dapat dimanfaatkan, beberapa diantaranya yaitu dapat membaca data gambar atau video dari file serta sebaliknya, menjalankan berbagai standar algoritma computer vision seperti line detection, edges detection, dan sebagainya (Sidharta, 2017).

2.14 Confusion Matrix

Confusion matrix atau *error matrix* adalah ringkasan hasil prediksi pada permasalah klasifikasi. Jumlah klasifikasi yang benar dan yang salah dikumpulkan dengan nilai hitung kemudian dipecah oleh setiap kelas, sehingga *confusion matrix* tidak hanya memberikan informasi kesalahan yang dibuat classifier tetapi juga jenis kesalahannya (Rusydi Umar, Imam Riadi, Purwono, 2020). Confusion matrix dapat dilihat dengan beberapa ketentuan sebagai berikut:

1. Positive (P): aktual bernilai positif. Negative (N): aktual bernilai negatif
2. True Positive (TP): aktual bernilai positif, dan diprediksi positif
3. True Negative (TN): aktual bernilai negative, dan diprediksi negatif
4. False Positive (FP): aktual bernilai negatif, tetapi diprediksi positif
5. False Negative (FN): aktual bernilai positif, tetapi diprediksi negatif

2.7.1. Precision

Nilai presisi dihitung dengan cara membagi total sampel positif yang diklasifikasikan dengan benar dengan total sampel positif yang diprediksi seperti pada Persamaan 2.1. Presisi tinggi menunjukan contoh berlabel positif memang positif (FP sedikit) (Rusydi Umar, Imam Riadi, Purwono, 2020).

$$Precision = \quad TP \quad TP + FP \tag{2.1}$$

1. High Recall, Low Precision adalah beberapa sampel positif yang dapat dikenali (FN rendah) tetapi terdapat banyak positif palsu.
2. Low Recall, High Precision adalah kehilangan banyak sampel positif (FN tinggi) namun yang diprediksi positif benar-benar positif (FP rendah).

2.7.2. Recall

Recall adalah rasio dari total keseluruhan sampel positif yang diklasifikasikan dengan benar kemudian dibagi dengan total sampel positif. High recall menunjukan bahwa kelas dideteksi dengan benar (FN sedikit). Recall dapat dihitung dengan Persamaan 2.2 (Rusydi Umar, Imam Riadi, Purwono, 2020).

$$Recall = \quad TP \quad TP + FN \tag{2.2}$$

2.7.3. F-Measure

F-Measure bertujuan untuk menghitung kombinasi dari presisi dan recall. *F-Measure* akan menggunakan harmonic mean dari presisi dan recall. Nilai F-Measure dihitung menggunakan Persamaan 2.3 (Rusydi Umar, Imam Riadi, Purwono, 2020).

$$F - Measure = 2\,x\,Precision * Recall \quad Precision + Recall \tag{2.3}$$

2.7.4. Mean Average Precision (mAP)

Mean average precision merupakan nilai rata-rata dari average precision (AP) yang membentuk metrik evaluasi untuk mengukur kinerja dari sebuah deteksi objek. Nilai AP didapatkan dari perhitungan precision pada persamaan 2.1 dan perhitungan recall pada persamaan 2.2, yang selanjutnya dilakukan perhitungan seperti pada persamaan 2.4 untuk mengetahui AP, lalu 2.5 untuk mengetahui mAP (Rusydi Umar, Imam Riadi, Purwono, 2020).

$$AP = \sum (recall_{n+1} - recall_{n}).\ Precision_{interp.}\ (recall_{n+1}) \tag{2.4}$$

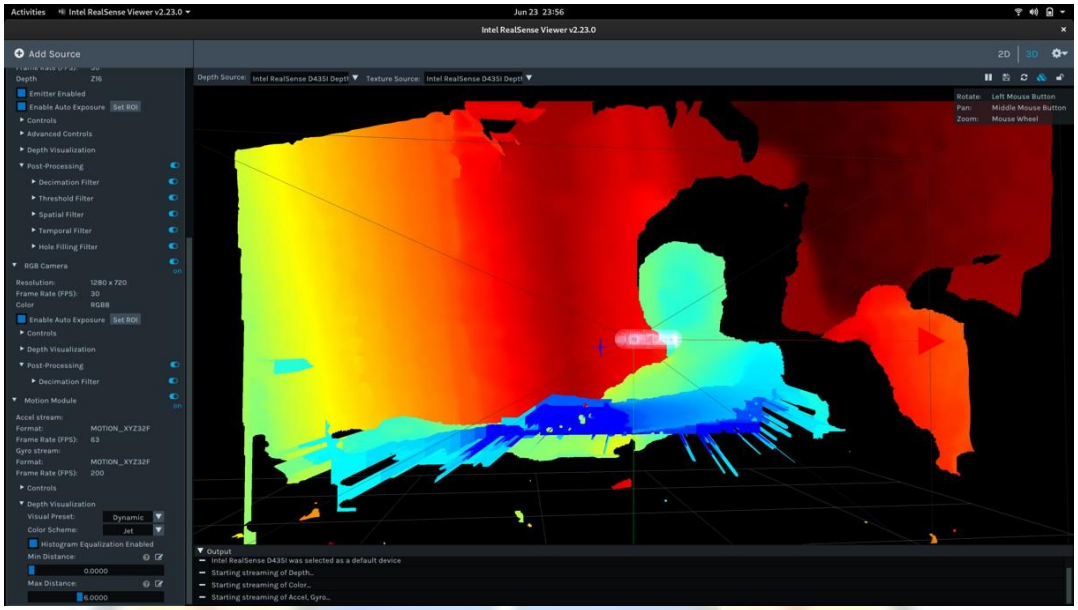
$$mAP = \quad i = 1 \quad N \quad AP(i) \quad N \quad x\,100\% \tag{2.5}$$

2.15 Intel RealSense Camera

Intel RealSense merupakan kamera yang memiliki 3 buah lensa, sebuah lensa 2D standar yang digunakan untuk mem-foto ataupun video, sebuah kamera infra-merah dan sebuah projector laser infra-merah. Infra-merah tersebut merupakan bagian yang membuat RealSense dapat melihat adanya jarak antara benda-benda, memisahkan benda-benda tersebut dari background di belakangnya dan akan mengenali benda, wajah, serta gerakan dibandingkan kamera biasa. Gambar 2.12 memperlihatkan kamera Intel Realsense.

Gambar 2.12 Kamera Intel RealSense

Teknologi RealSense pada dasarnya terdiri dari prosesor untuk pemrosesan gambar, modul untuk membentuk gambar kedalaman, modul untuk melacak gerakan, dan kamera kedalaman. Kamera-kamera ini mengandalkan teknologi pemindaian mendalam, yang memungkinkan komputer melihat objek dengan cara yang sama seperti manusia (TADIC, et al., 2019).



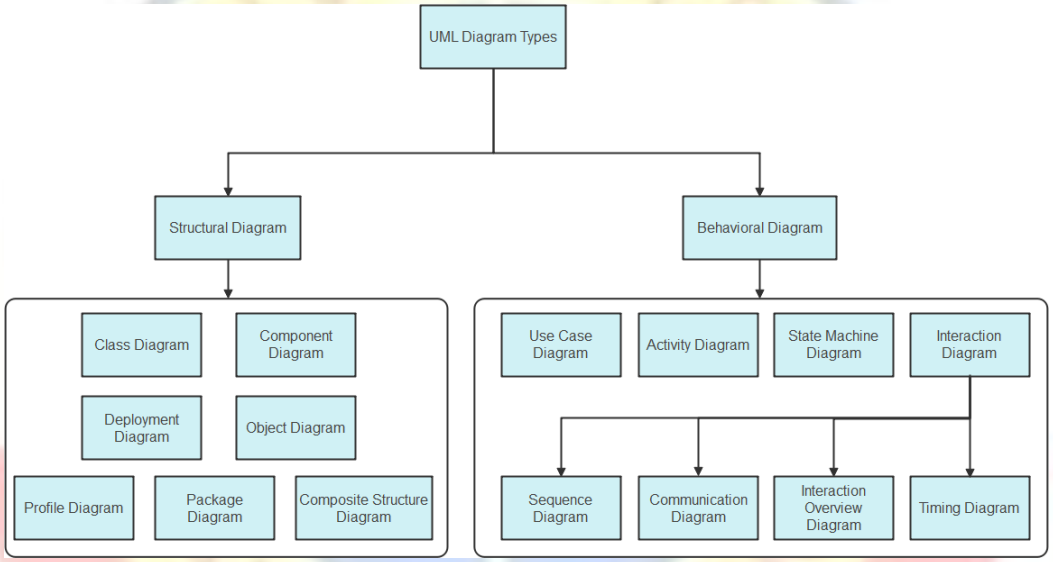
Gambar 2.13 Viewer Intel RealSense

Selain perangkat keras lengkap, terdapat pula platform perangkat lunak SDK (Software Development Kit) sumber terbuka yang sesuai yang disebut librealsense. Platform perangkat lunak ini menyediakan dukungan perangkat lunak sederhana untuk kamera seri D400, yang memungkinkan pengguna untuk menggunakan kamera-kamera tersebut. Platform perangkat lunak ini mendukung sistem ROS (Robot Operating System), C/C++, Matlab, dan bahasa pemrograman Python, yang dapat digunakan untuk mengembangkan aplikasi yang sesuai.

Dalam penggunaan kamera Intel Realsense, disediakan juga aplikasi Viewer untuk kamera tersebut seperti pada Gambar 2.13, dimana pengguna bisa melakukan konfigurasi terhadap kamera.

2.16 UML (Unified Modeling Language)

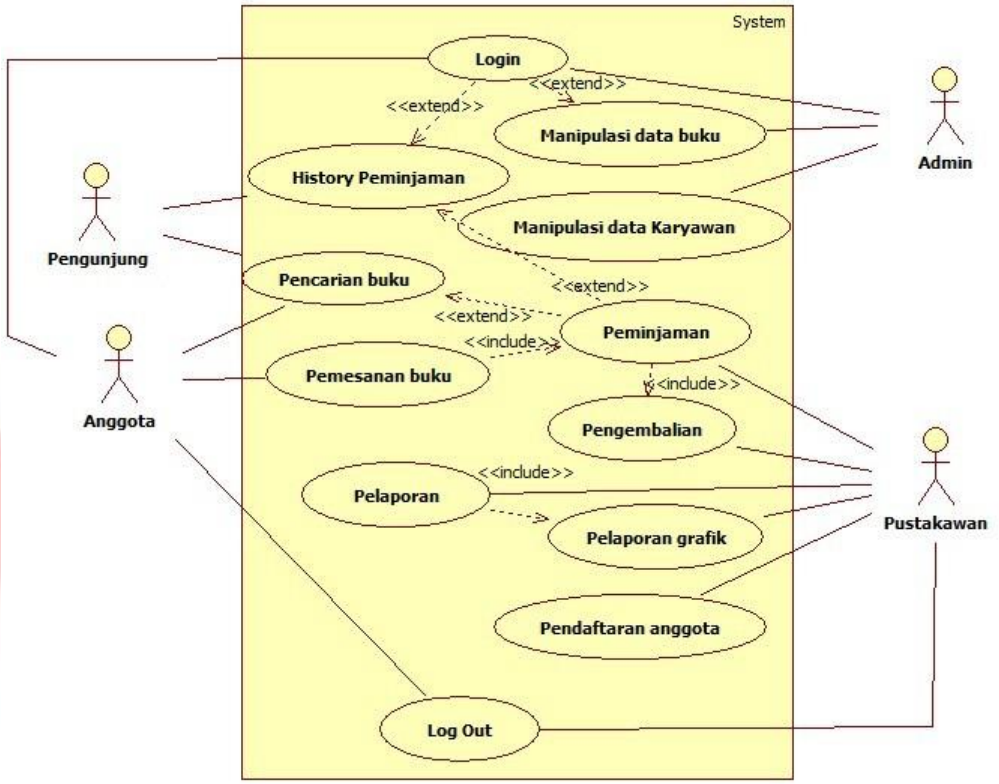
UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (ObjectOriented) (Dharwiyanti & Wahono, 2023). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.



Gambar 2.14 Diagram UML

2.16.1 Use Case Diagram

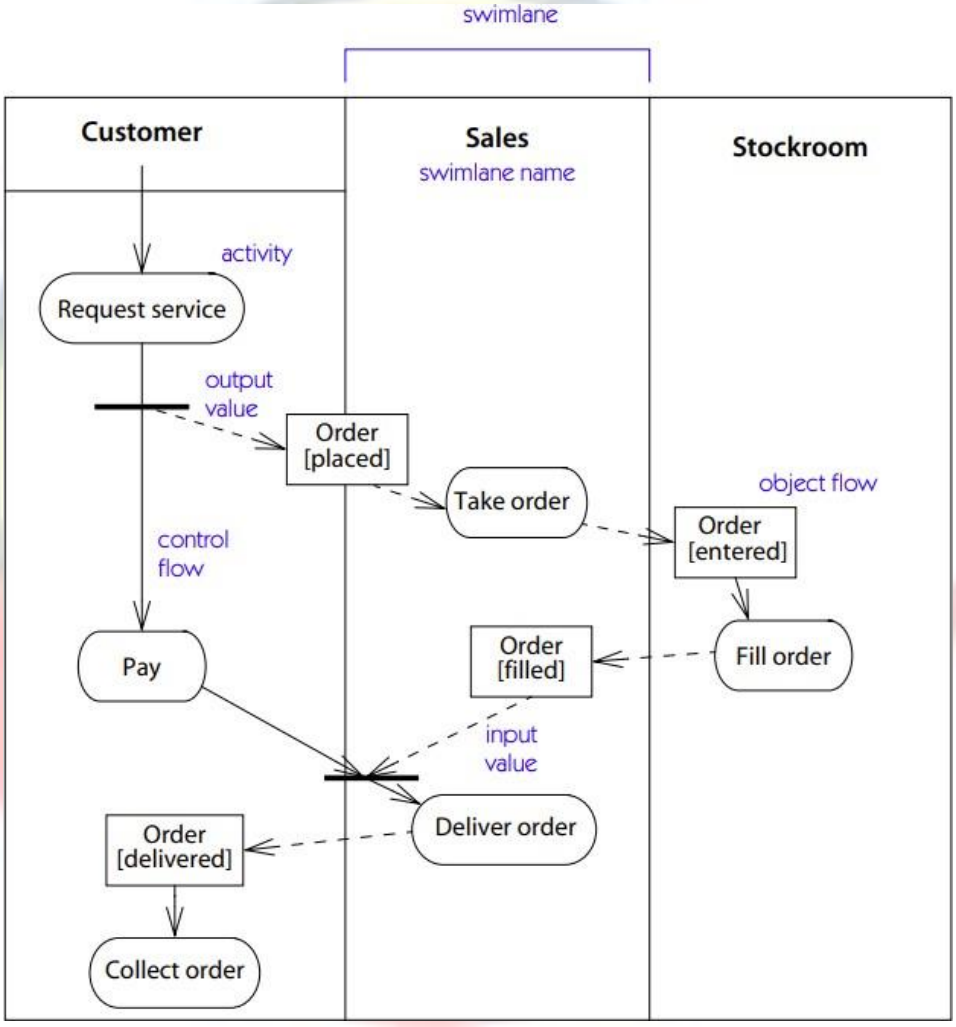
Use case menunjukkan fungsi sistem dari sudut pandang pengguna dan menentukan apa yang akan diproses sistem dan bagian- bagiannya. Use case bekerja dengan skenario, yang menunjukkan urutan atau langkah-langkah tindakan pengguna terhadap sistem dan sebaliknya. Fungsi sistem, cara pengguna berinteraksi dengannya, dan hubungan antara pengguna dengannya diidentifikasi oleh use case (Setiyani, 2021). Gambar 2.15 merupakan contoh dari *use case diagram* pada sistem perpustakaan.



Gambar 2.15 Contoh Use Case Diagram

2.16.2 Activity Diagram

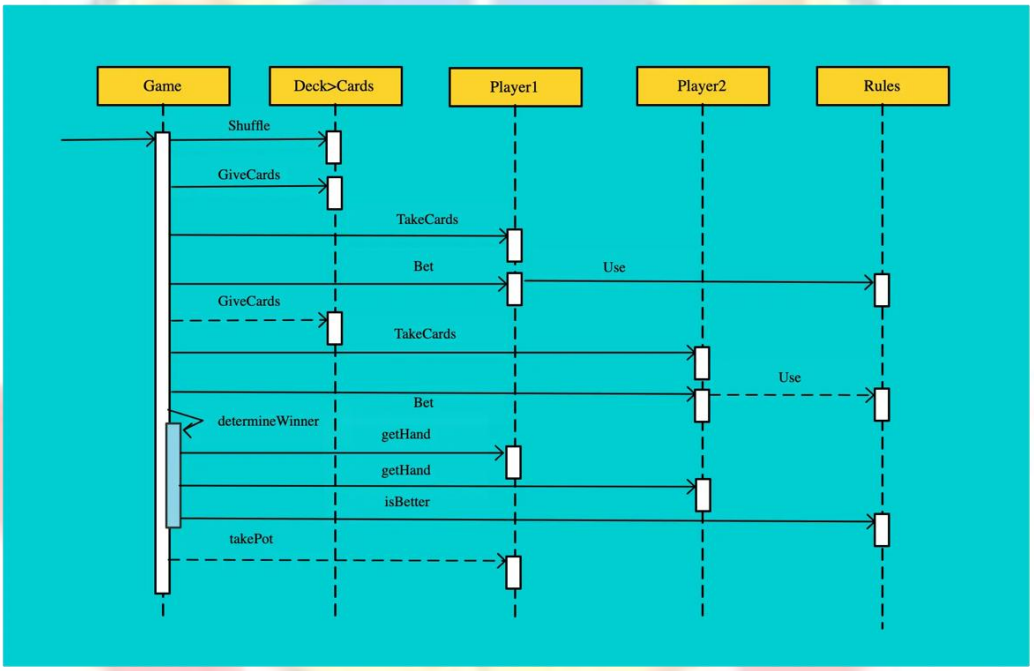
Activity diagram merupakan jenis pemodelan yang mendeskripsikan alur kerja sebuah objek atau sistem. Mereka menunjukkan proses yang digunakan untuk memproses usecase dari awal hingga akhir, dan setiap aktivitas diberi label berdasarkan fungsinya. Gambar 2.16 merupakan contoh dari activity diagram.



Gambar 2.16 Contoh Activity Diagram

2.16.3 Sequence Diagram

Sequence Diagram merupakan jenis pemodelan digunakan dalam menjelaskan karakter pada sebuah scenario serta menunjukkan bagaimana pengguna berinteraksi dengan sistem, termasuk perintah saat berinteraksi. Mereka juga menunjukkan komunikasi di antara objek dan interaksi mereka. Pada eksekusi, semua pesan ditampilkan dalam urutan. Sequence diagram menunjukkan perilaku objek yang terdapat dalam usecase serta menampilkan pesan yang dikirimkan dan diterima antara objek dan juga waktu hidup objek. Gambar 2.17 merupakan contoh dari sequence diagram.



Gambar 2.17 Contoh Sequence Diagram

2.17 Blok Diagram

Blok diagram yaitu suatu gambaran dasar dari rangkaian sistem yang akan dirancang, dimana setiap bagian blok diagram memiliki fungsi masing-masing.

Blok yang memberikan diagram blok namanya mewakili elemen yang berbeda dalam suatu sistem. Garis dan panah menunjukkan hubungan antara blok-blok itu. Elemen visual ini memberikan gambaran fungsional tingkat tinggi dari sistem yang mudah dicerna dan dipahami.

Membuat diagram blok membantu semua orang yang terlibat dalam proyek memahami dan memvisualisasikan dengan tepat apa yang diperlukan agar sesuatu berfungsi secara keseluruhan. Mereka menciptakan pemahaman yang koheren tentang elemen-elemen yang diperlukan untuk menghubungkan bersama untuk menciptakan hasil akhir yang diinginkan. Dengan cara ini, mereka menjaga semua orang di tim di halaman yang sama dan bekerja menuju tujuan bersama (Miro, 2024).

2.18 Python

Python adalah bahasa pemrograman yang bersifat freeware atau bebas, yang berarti tidak ada batasan dalam penyalinan atau distribusinya. Python dilengkapi dengan source code, debugger, profiler, antarmuka, fungsi sistem, GUI (Graphics User Interface), dan baris datanya. Python menjadi bahasa resmi yang terintegrasi dalam Raspberry Pi. Nama Raspberry Pi sendiri terinspirasi dari kata "Python", dan dikatakan bahwa Python adalah bahasa yang secara alami terkait dengan Raspberry Pi. Logo Python dapat dilihat pada Gambar 2.18.

Gambar 2.18 Logo Python

Python mendukung berbagai paradigma pemrograman, terutama namun tidak terbatas pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang dimiliki oleh Python adalah sebagai bahasa pemrograman dinamis dengan manajemen memori otomatis. Seperti halnya bahasa pemrograman dinamis lainnya, Python sering digunakan sebagai bahasa skrip, meskipun pada kenyataannya, penggunaannya lebih luas dan mencakup berbagai konteks pemanfaatan yang tidak terbatas pada bahasa skrip. Python dapat digunakan untuk berbagai tujuan dalam pengembangan perangkat lunak dan dapat dijalankan di berbagai platform sistem operasi.

2.19 PIP

PIP atau disebut *Package Installer for Python* adalah alat yang digunakan untuk menginstal berbagai paket dalam Python. Paket Python adalah sekumpulan kode Python yang dapat diimpor ke dalam program Python. PIP adalah alat yang sering digunakan dalam pengembangan Python untuk manajemen paket, seperti menambahkan, mengupgrade, dan menghapus paket Python dengan mudah. PIP telah tersedia dalam Python versi 2.7.9 ke atas atau Python 3.4.

PIP menawarkan *one-click solution* untuk *library* python sehingga memungkinkan pengembang dalam menggunakan kode orang lain untuk mempercepat pengembangan aplikasi (Liang, et al., 2021).

2.20 Flask

Flask merupakan sebuah kerangka kerja web yang dikodekan dengan menggunakan bahasa pemrograman Python dan termasuk dalam kategori microframework. Flask berperan sebagai kerangka kerja untuk pengembangan aplikasi web dan mengatur tampilan dari suatu situs web. Dengan memanfaatkan Flask dan bahasa pemrograman Python, pengembang dapat membuat situs web yang terstruktur dan mengelola perilaku suatu situs web dengan lebih mudah. Logo Flask Framework dapat ditemukan pada Gambar 2.19.

Gambar 2.19 Logo Flask

Flask termasuk dalam kategori microframework karena tidak memerlukan alat atau pustaka khusus dalam penggunaannya. Sebagian besar fungsi dan komponen umum, seperti validasi formulir, database, dan lainnya, tidak terpasang secara default di Flask. Hal ini disebabkan oleh fakta bahwa fungsi-fungsi dan komponen-komponen tersebut telah disediakan oleh pihak ketiga, dan Flask dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seolah-olah diimplementasikan oleh Flask itu sendiri.

Meskipun disebut sebagai microframework, hal ini tidak berarti Flask memiliki keterbatasan dalam hal fungsionalitas. Istilah "microframework" pada Flask merujuk pada tujuan untuk membuat inti aplikasi sebersih mungkin, tetapi tetap mudah untuk ditambahkan. Oleh karena itu, fleksibilitas dan skalabilitas Flask dapat dianggap tinggi dibandingkan dengan beberapa framework lainnya.

Flask, sebagai web framework, ditulis dalam bahasa Python. Oleh karena itu, sebelum menggunakan Flask, seorang pengembang web perlu menginstal Python di perangkatnya. Oleh karena itu, seorang pengembang web yang ingin menggunakan Flask untuk pengembangan web perlu memiliki pengetahuan dasar tentang bahasa pemrograman Python.

Dalam melakukan instalasi Flask pada suatu perangkat, dibutuhkan PIP yang umumnya sudah terinstal pada Python versi 3.4 ke atas. PIP, sebagai sistem manajemen paket, digunakan untuk mengelola dan menginstal paket yang berisi modul-modul Python. Penggunaan PIP pada instalasi Flask memungkinkan untuk mengunduh dan memasang semua komponen yang diperlukan dengan satu perintah. Hal ini karena Flask ditulis dan dikembangkan menggunakan bahasa pemrograman Python serta modul-modulnya.

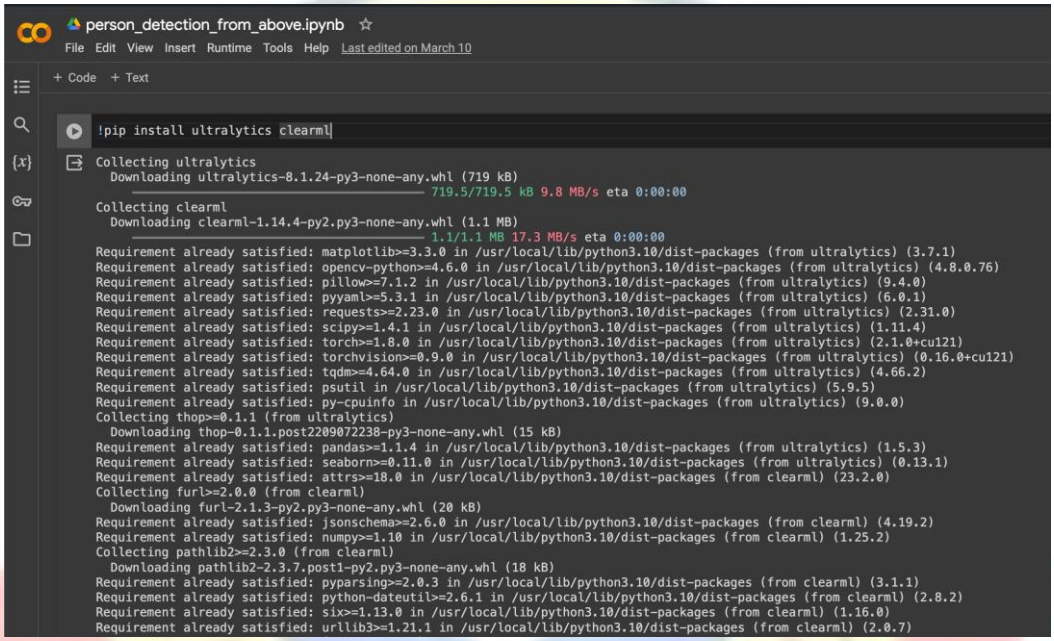
2.21 Web Browser

Web Browser adalah perangkat lunak yang digunakan untuk menampilkan informasi dari web server. Browser ini dikembangkan dengan menggunakan Graphic User Interface (GUI), memungkinkan pengguna untuk dengan mudah melakukan point dan click untuk berpindah antar dokumen. Sebagai contoh, Lynx adalah salah satu browser yang masih menggunakan mode teks, sehingga tidak dapat menampilkan gambar. Lynx biasanya ditemui pada lingkungan DOS dan Unix. Namun, seiring perkembangan teknologi, web browser saat ini umumnya menggunakan GUI. Beberapa contoh browser GUI populer antara lain Mozilla Firefox, Internet Explorer, Opera, dan Google Chrome seperti pada Gambar 2.20. Saat ini, browser bersaing untuk mendapatkan pengguna dengan mendekati standar dokumen HTML yang direkomendasikan oleh W3C.

Gambar 2.20 Macam-macam Web Browser

2.22 Google Colaboratory

Google Colaboratory lebih umum disebut sebagai "Google Colab" atau hanya "Colab" adalah proyek penelitian untuk membuat prototipe model pembelajaran mesin pada opsi perangkat keras yang kuat seperti GPU dan TPU. Ini menyediakan lingkungan notebook Jupyter tanpa server untuk pengembangan interaktif. Google Colab gratis digunakan seperti produk G Suite lainnya (Bisong, 2019).



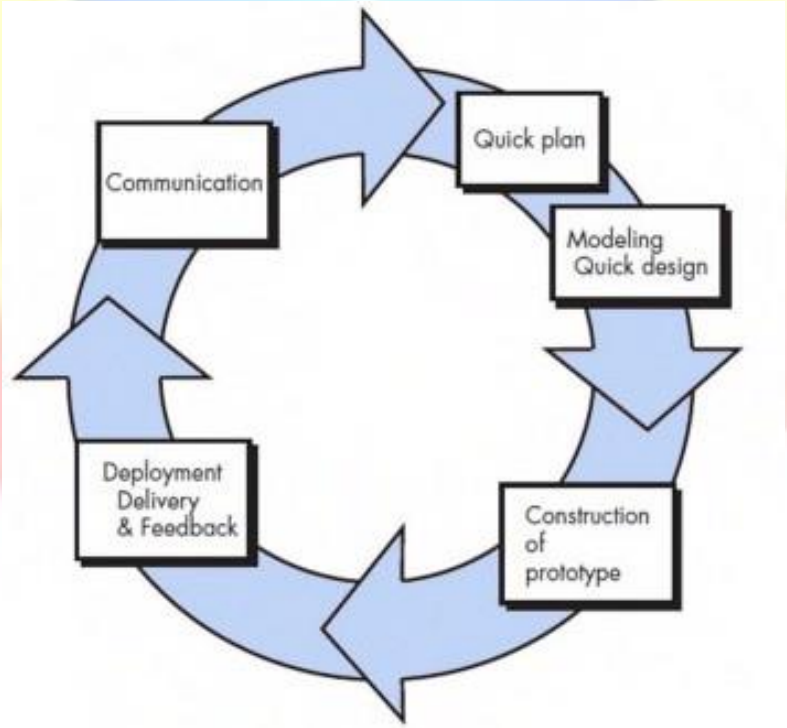
Gambar 2.21 Tampilan Google Colaboratory

Google Colaboratory berperan seperti komputer, sehingga bisa dilakukan penginstalan terhadap aplikasi-aplikasi yang dibutuhkan dalam proyek penelitian. Gambar 2.21 merupakan contoh tampilan google colab yang sedang menginstall beberapa *library python*.

BAB III
ANALISIS DAN PERANCANGAN

Bab ini menjelaskan mengenai perancangan sistem menggunakan salah satu metodologi pengembangan perangkat lunak, yaitu *prototype*. *Prototype* merupakan metodologi yang digunakan untuk mengembangkan versi awal dari sebuah produk perangkat lunak. Versi awal ini digunakan untuk mendemostrasikan fungsionalitasnya, mengumpulkan masukan dan memperbaiki kebutuhan-kebutuhan produk perangkat lunak tersebut sebelum pengembangan skala penuh.

Tahapan-tahapan pada pengembangan perangkat lunak menggunakan metodologi *prototype* terdiri dari beberapa tahapan yaitu *communication, quick plan, modeling quick design, construction of prototype, deployment delivery & feedback*, lalu kembali ke *communication* (Pressman, 2010). Gambar 3.1 adalah tahapan dari metodologi *prototype*.



Gambar 3.1 Model Prototype

3.1. Analisis Kebutuhan (*Communication*)

Pada Tahapan analisis kebutuhan dijelaskan kebutuhan-kebutuhan yang ada pada sistem pada penelitian ini. Kebutuhan tersebut terbagi menjadi kebutuhan perangkat lunak dan kebutuhan perangkat keras.

3.1.1 Kebutuhan Perangkat Keras

Kebutuhan atau spesifikasi perangkat keras memiliki pengaruh terhadap perolehan FPS serta akurasi pendeteksian pada saat pengujian. Perangkat dengan spesifikasi CPU diatas 3.0 GHz, RAM minimum 16 GB, dan GPU yang lebih dari 4GB dapat menjalankan model penelitian secara real-time dengan baik. (Putra, et al., 2023)

Pada penelitian ini, perangkat yang digunakan untuk melakukan uji coba memiliki spesifikasi sebagai berikut:

3. Prosesor Apple M1 Pro
4. RAM 16 GB
5. 14 Core GPU

3.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem deteksi objek manusia dijelaskan sebagai berikut:

1. Mac OS Ventura 13.6 dibutuhkan sebagai sistem operasi pada laptop yang digunakan untuk menuliskan dokumentasi laporan penelitian serta menjalankan sistem deteksi objek manusia.
2. Visual Studio Code merupakan aplikasi editor untuk kode sistem deteksi objek manusia
3. Google Colaboratory merupakan platform bahasa pemrograman python berbasis web yang digunakan untuk membuat bobot pada sistem deteksi objek manusia

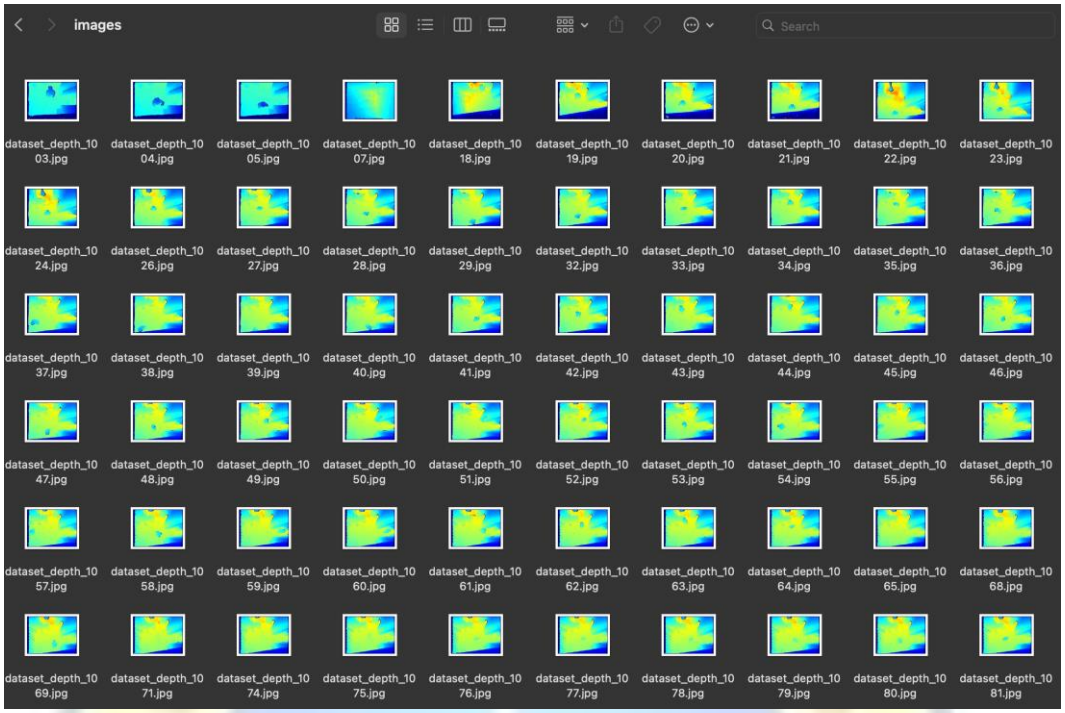
3.2. Dataset dan Pre-Processing Data

Langkah awal dalam mendeteksi objek manusi adalah mempersiapkan data berupa frame dengan objek manusia dan frame tanpa objek manusia dalam format JPG maupun JPEG. Frame yang dijadikan dataset pada penelitian ini berasal dari gambar hasil perekaman yang di lakukan pada tempat penelitian menggunakan kamera depth sensor. Input dari rancangan sistem yang digunakan berupa gambar dengan resolusi yang sama yaitu 640x480 piksel.

Tabel 3.1 Dataset Pelatihan dan Validasi

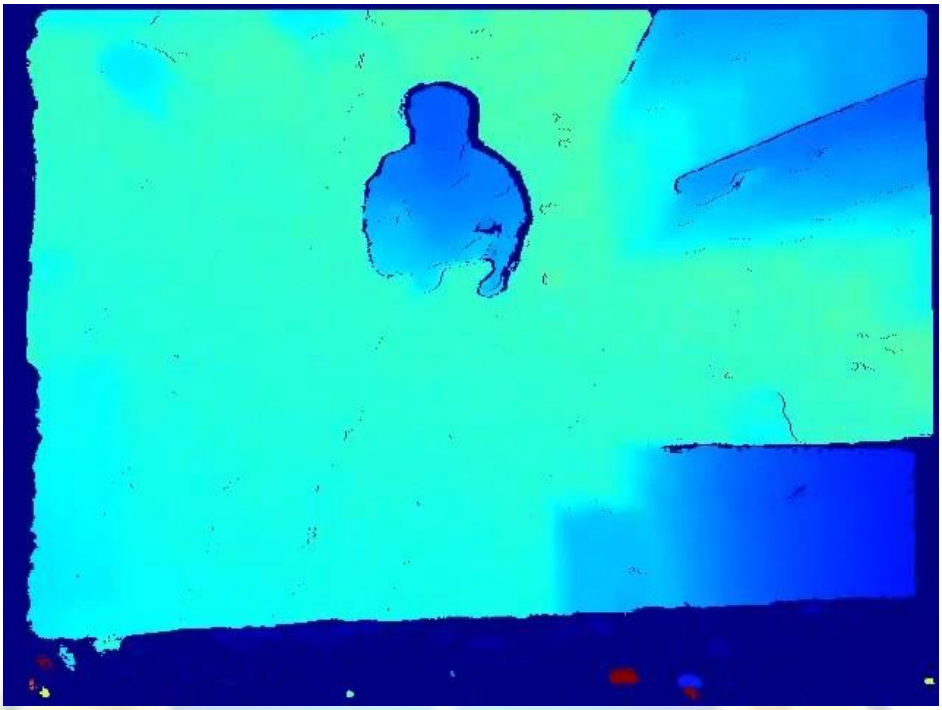
Tipe	**Objek Manusia**	**Background**	**Total**
Pelatihan	536	184	720
Validasi	133	47	180
Total			900

Seperti pada Tabel 3.1, jumlah total data yang digunakan peneliti dalam pelatihan dan validasi model yaitu sebanyak 900 gambar. Data tersebut dibagi menjadi dua bagian yaitu dataset training dan dataset validasi, dimana dataset training menggunakan 720 gambar (536 berobjek manusia, 184 background) dan dataset validasi 180 gambar (133 berobjek manusia, 47 background). Gambar 3.2 merupakan contoh dari dataset yang akan digunakan pada tahapan pelatihan dan validasi.



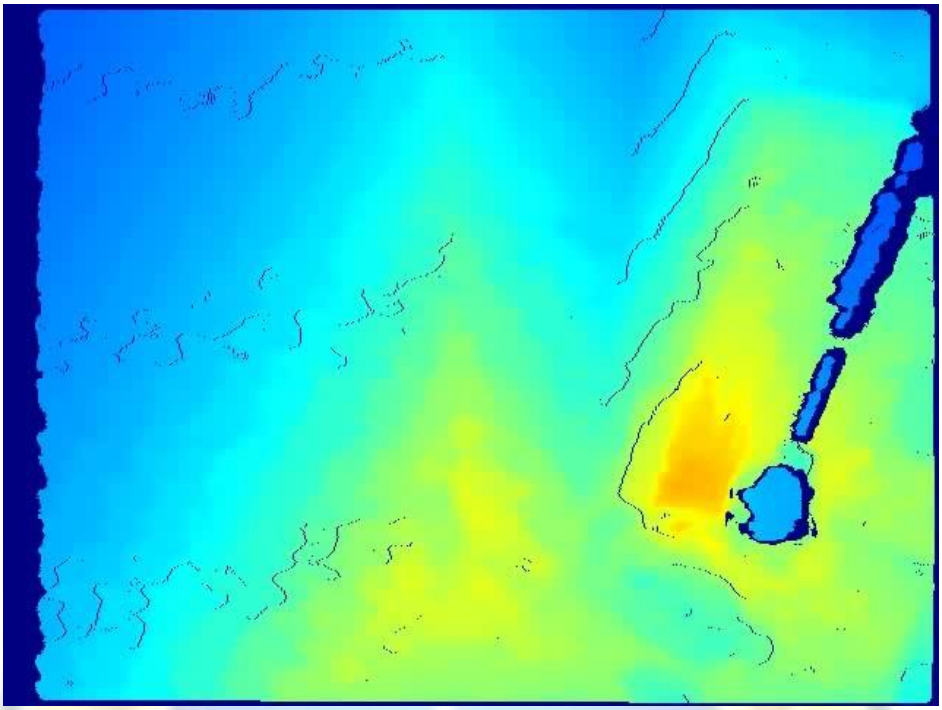
Gambar 3.2 Dataset untuk pelatihan dan validasi

Gambar 3.2 merupakan contoh dataset untuk pelatihan dan validasi. Contoh frame kamera depth sensor yang memiliki objek manusia berada pada Gambar 3.3 dan Gambar 3.4, sedangkan Gambar 3.5 dan Gambar 3.6 merupakan contoh frame kamera depth sensor yang tidak memiliki objek manusia.

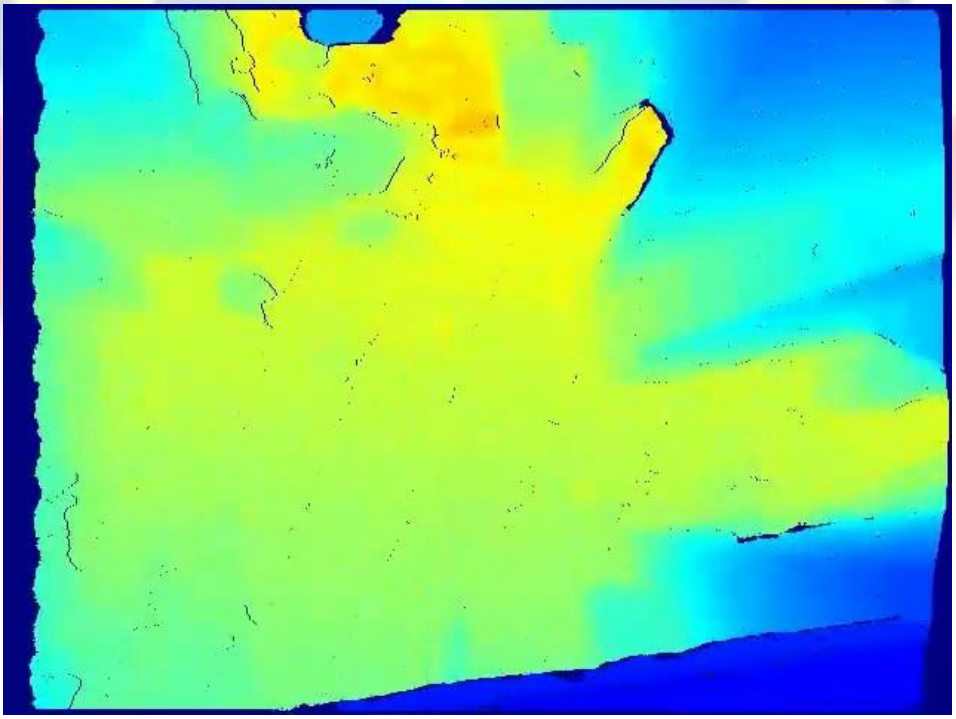


Gambar 3.3 Frame Kamera Depth Sensor dengan Objek Manusia

Gambar 3.4 Frame Kamera Depth Sensor dengan Objek Manusia (2)

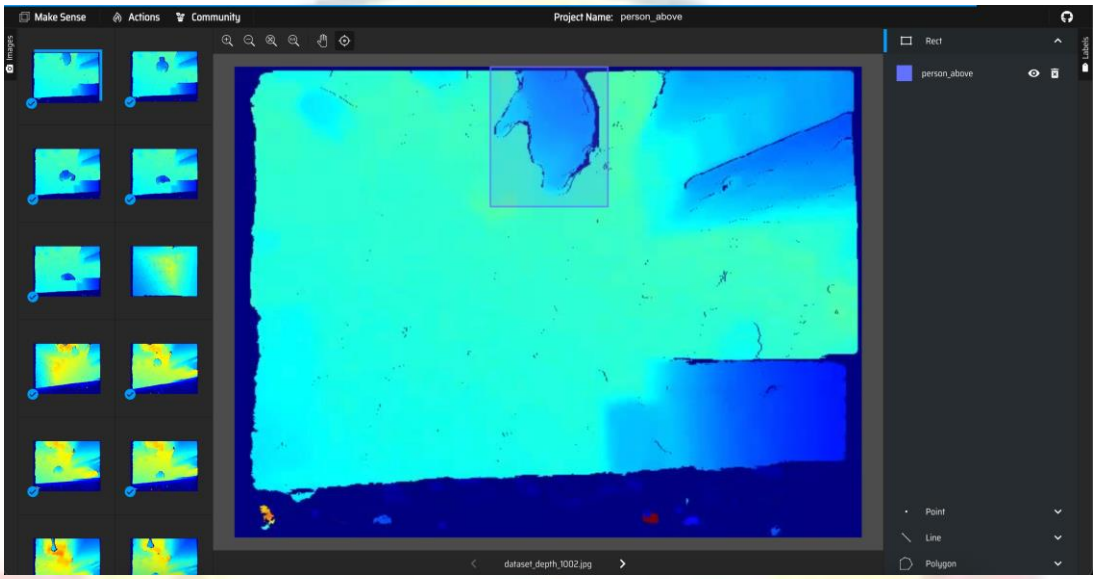


Gambar 3.5 Frame Kamera Depth Sensor tanpa Objek Manusia



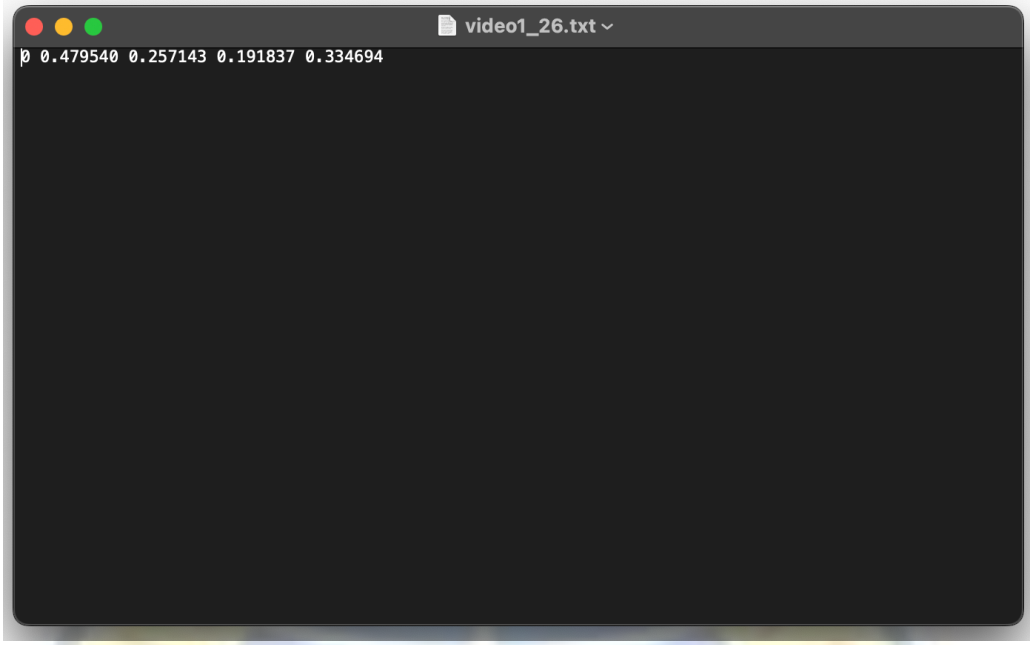
Gambar 3.6 Frame Kamera Depth Sensor tanpa Objek Manusia (2)

Tiap frame yang digunakan pada penelitian melalui tahapan pelabelan atau *labeling.* Tahapan pelabelan dilakukan secara manual melalui alat bernama makesense.ai. Total frame yang melalui tahapan pelabelan yaitu sebanyak 900 gambar. Tiap tiap objek manusia yang berada pada frame diberikan kotak dan juga diberi *class* person_above seperti pada Gambar 3.7.



Gambar 3.7 Frame yang sudah dilabeli class person_above

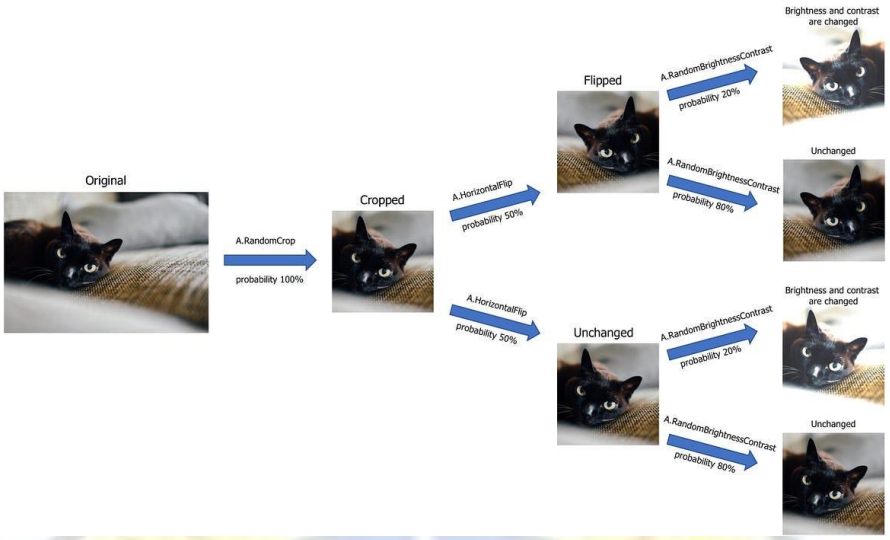
Setelah tahapan pelabelan selesai, hasil dari tahapan tersebut kemudian disimpan dalam format yang sesuai dengan format pelabelan gambar YOLO. Setiap frame yang memiliki objek manusia didalamnya akan mendapatkan file berformat .txt dengan nama yang sama. File .txt tersebut berisi informasi berupa kelas beserta koordinat dari tiap objek manusia yang ada pada frame.



Gambar 3.8 Isi file .txt hasil pelabelan pada frame

Gambar 3.8 merupakan contoh isi dari file .txt hasil pelabelan pada frame. Tiap baris pada file tersebut terdiri dari index class, koordinat x, koordinat y, lebar dan tinggi dari *bounding box* objek tersebut.

YOLOv8 menyediakan *data preprocessing* dan *data augmentation*-nya sendiri yang diterapkan secara otomatis pada tiap dataset pada tahap pelatihan model.



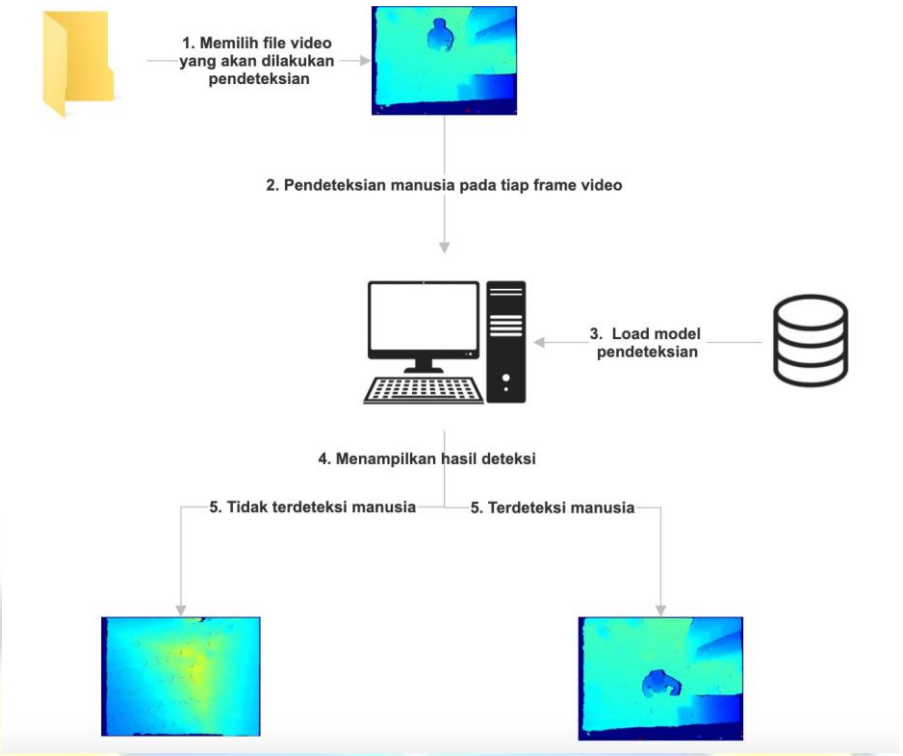
Gambar 3.9 Contoh data augmentation

Gambar 3.9 merupakan contoh dari data augmentation. *Data preprocessing* dan *Data augmentation* yang diterapkan oleh YOLOv8 yaitu (Ultralytics, 2024):

1. Mosaic
2. MixUp
3. RandomPerspective
4. RandomHSV
5. RandomFlip
6. LetterBox
7. CopyPaste
8. Albumentations
9. CenterCrop

3.3. Pemodelan Sistem

Pada tahap pemodelan sistem diilustrasikan alur kerja sistem dari mulai pemilihan data uji dari direktori, hingga sistem mendeteksi objek manusia. Pada Gambar 3.10 diilustrasikan gambaran kerja sistem secara keseluruhan.



Gambar 3.10 Pemodelan Sistem

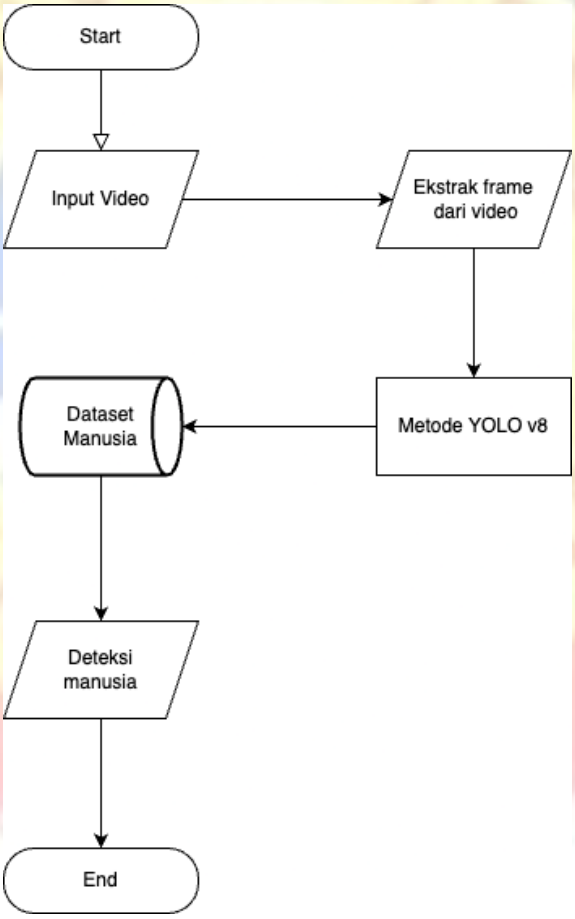
Dalam Gambar 3.10 dari permodelan sistem, terdapat 5 tahapan proses yang dijalankan oleh sistem yang dibangun untuk mendeteksi objek manusia. Berikut adalah tahapan-tahapan tersebut:

10. Langkah pertama yang dilakukan adalah memilih *file* data uji berupa video pada direktori.
11. Langkah kedua adalah proses deteksi objek manusia dengan menggunakan algoritma YOLOv8.
12. Langkah ketiga adalah *load* model hasil training yang berisi bobot dan bias dataset, dan dilakukan proses pencocokan dari data uji dengan model sehingga menghasilkan kelas objek yang dikenali berdasarkan dataset.
13. Langkah keempat adalah sistem menampilkan hasil deteksi objek berupa *bounding box*, dan label pada data uji.
14. Langkah kelima keluaran sistem menampilkan hasil deteksi objek manusia.

3.4. Deteksi Objek Manusia dengan YOLOv8

Flowchart yang dibuat pada Gambar 3.11 merepresentasikan cara kerja sistem untuk deteksi objek manusia. Proses sistem deteksi objek manusia terdapat beberapa tahapan yaitu tahap pertama masukan sistem berupa file video.

Kemudian tahap kedua dilakukan proses metode Algoritma YOLOv8 untuk menghasilkan nilai bobot secara konvolusional pada frame. Selanjutnya tahap ketiga dilakukan proses pencocokan nilai bobot data uji dengan nilai bobot dataset. Tahap keempat keluaran sistem yang dihasilkan dapat mendeteksi objek manusia.



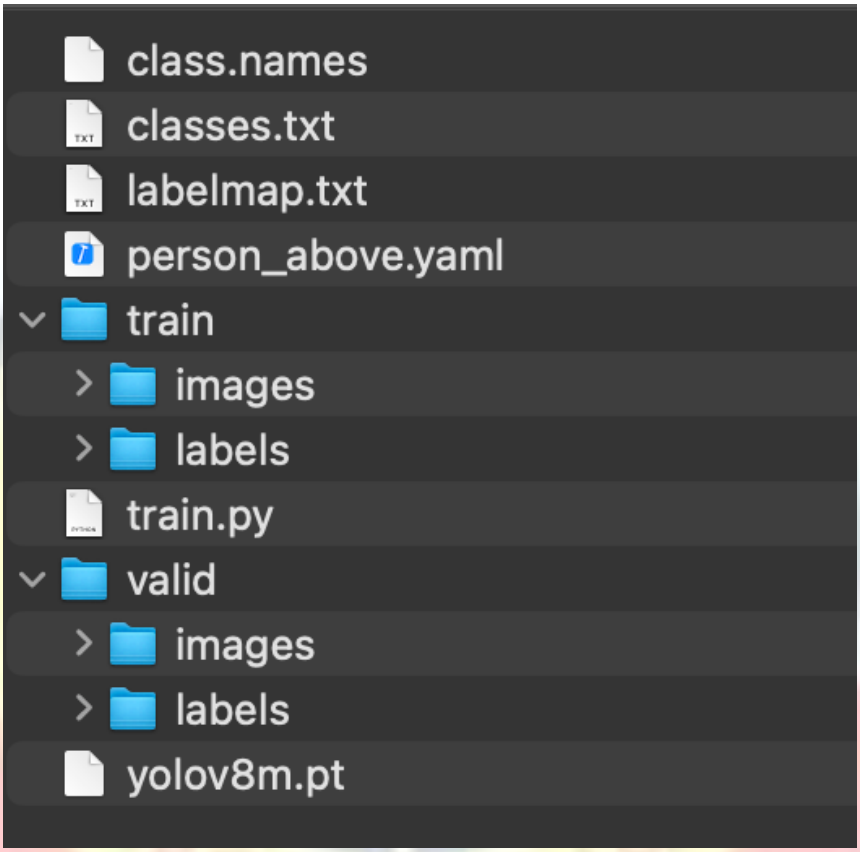
Gambar 3.11 *Flowchart*

Proses algoritma YOLOv8 pada sistem deteksi objek manusia mencakup beberapa tahapan utama, yang dapat disajikan sebagai berikut:

1. Preprocessing Data: Langkah pertama dalam proses adalah mempersiapkan data pelatihan. Ini melibatkan pembacaan dan pemrosesan gambar yang akan digunakan sebagai input untuk pelatihan model deteksi objek manusia.
2. Inisialisasi Model: Model YOLOv8 akan diinisialisasi dengan parameter awal. Ini melibatkan konfigurasi arsitektur model, termasuk jumlah lapisan, ukuran kernel, dan konfigurasi lainnya.
3. Pelatihan Model: Dataset testing akan digunakan untuk melatih model YOLOv8. Ini melibatkan proses iteratif di mana model diberi masukan gambar dan label yang sesuai, dan kemudian menyesuaikan bobotnya agar sesuai dengan data pelatihan.
4. Validasi Model: Setelah pelatihan, model akan divalidasi menggunakan dataset validasi. Tujuan dari langkah ini adalah untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya dan mendeteksi apakah terjadi *overfitting* atau *underfitting*.
5. Fine-Tuning: Jika diperlukan, model dapat disempurnakan lebih lanjut melalui proses fine-tuning. Ini mungkin melibatkan penyesuaian hyperparameter, penambahan lapisan tambahan, atau teknik peningkatan kinerja lainnya.
6. Evaluasi Model: Setelah proses fine-tuning, model akan dievaluasi menggunakan data pengujian untuk mengukur kinerjanya secara keseluruhan. Ini melibatkan penggunaan metrik evaluasi seperti *Precision*, *Recall*, dan *F1-Measure* untuk mengukur seberapa baik model dapat mendeteksi objek manusia dalam gambar.
7. Implementasi: Setelah model dianggap memadai, langkah terakhir adalah mengimplementasikannya dalam sistem deteksi objek manusia. Ini melibatkan integrasi model ke dalam aplikasi atau sistem yang lebih luas untuk digunakan dalam situasi nyata.

3.4.1. Pelatihan Model

Proses pelatihan model YOLOv8 dilakukan pada Google Colaboratory. Gambar 3.12 merupakan list file dan folder yang perlu disiapkan untuk nantinya diupload ke Google Colaboratory.



Gambar 3.12 File dan Folder untuk Pelatihan

training > ! person_above.yaml

```yaml
train: train/images
val: valid/images

nc: 1

# class names
names: ['person_above']

```

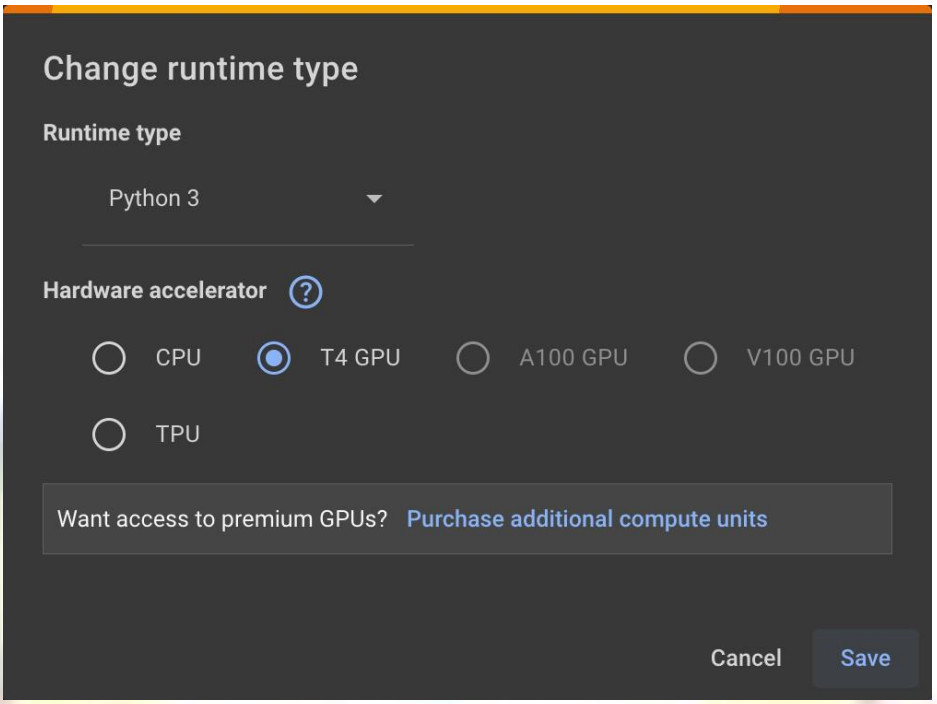
Gambar 3.13 Isi file person_above.yaml

File class.names, classes.txt dan labelmap.txt berisi kelas yang akan diidentifikasi pada model, yaitu person_above. Folder train dan valid merupakan kumpulan dataset yang sudah persiapkan. Masing-masing dataset tersebut akan digunakan untuk proses pelatihan dan validasi. File yolov8m.pt merupakan base model yang digunakan untuk melatih YOLOv8 terhadap objek manusia, dan file person_above.yaml seperti pada Gambar 3.13 merupakan file konfigurasi untuk pelatihan model.

```python
from ultralytics import YOLO

task = Task.init(project_name='person_above', task_name='person_above_colab')

model = YOLO('yolov8m.pt')

results = model.train(
    data = '/content/training/person_above.yaml',
    imgsz=640,
    epochs=150,
    batch=20,
    name='person_above'
)
```

Gambar 3.14 Kode Pelatihan Model Pada Google Colab

Gambar 3.14 merupakan potongan kode untuk pelatihan model pada Google Colaboratory. Konfigurasi yang digunakan untuk pelatihan model yaitu gambar dengan resolusi lebar 640 dengan tinggi menyesuaikan, total epochs 150 dan batch 20.



Gambar 3.15 Resource *Google Colaboratory*

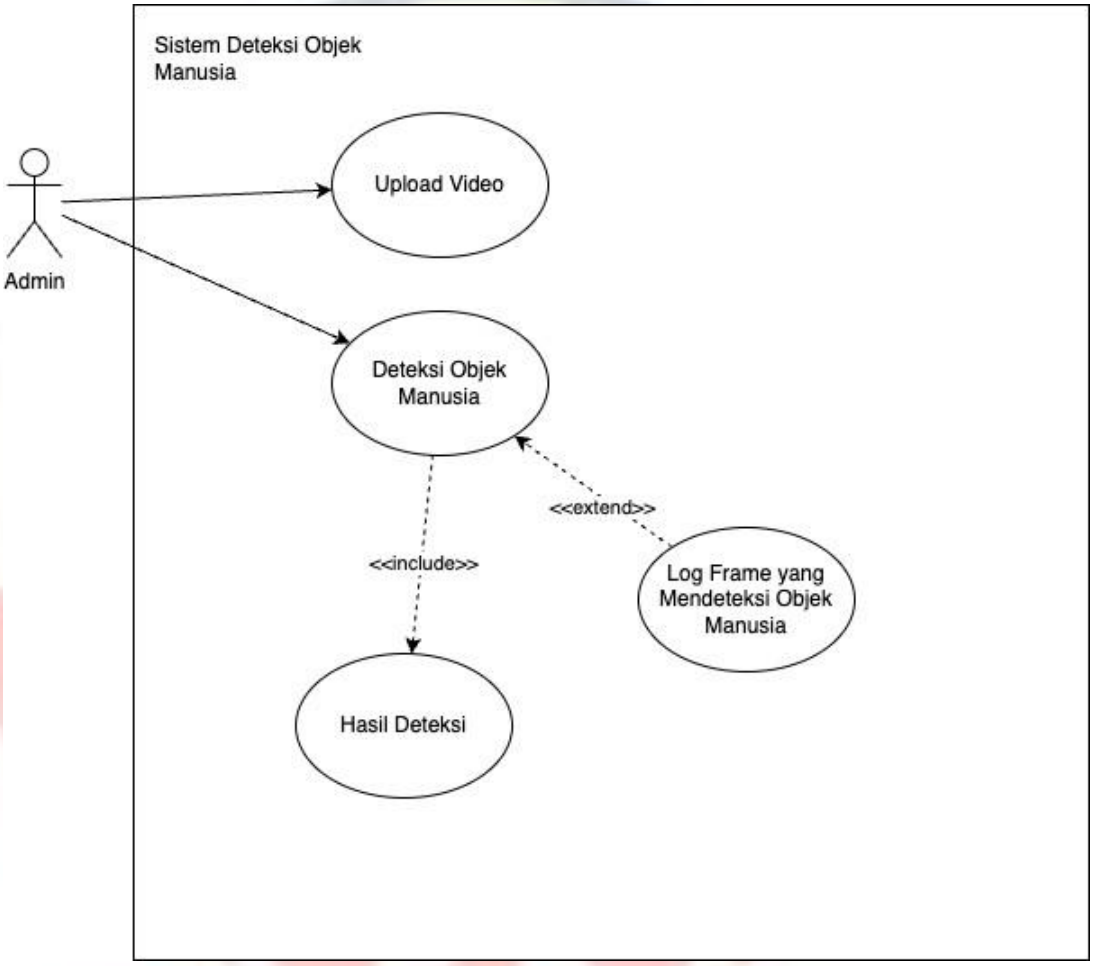
Gambar 3.15 merupakan *resource* yang digunakan untuk melakukan pelatihan pada Google Colaboratory, yaitu Python 3 dengan Hardware berupa T4 GPU, RAM 12 GB, serta penyimpanan sebesar 78 GB.

3.5. Quick Plan & Modelling Quick Design

Dalam perancangan sistem ini, akan dipertimbangkan aliran kerja sistem yang akan dikembangkan, serta rancangan aktor-aktor dan proses-proses yang akan berinteraksi dalam aplikasi tersebut dengan memanfaatkan *Use Case Diagram, Activity Diagram, Sequence Diagram,* dan *Blok Diagram.*

3.5.1. Use Case Diagram

Diagram Kasus Pengguna (*Use Case Diagram*) adalah representasi grafis yang dapat menggambarkan interaksi antara pengguna dengan sistem. Diagram Kasus Pengguna ini memberikan penjelasan mengenai fungsi-fungsi yang dilakukan oleh sistem. Use Case Diagram pada sistem deteksi objek manusia tersaji pada Gambar 3.8.



Gambar 3.16 *Use Case Diagram*

Pada Gambar 3.16 *use case diagram* menjelaskan hubungan admin dengan sistem yang telah dibangun. Skenario-skenario berikut ini menjelaskan tiap use case yang ada pada diagram tersebut:

1. *Use Case Diagram* Upload Video

Kegiatan atau interaksi antara admin dengan sistem dalam mengupload video dapat dilihat pada Tabel 3.2 berikut.

Tabel 3.2 Skenario Use Case Upload Video

Identifikasi	
Nomor Use Case	01
Nama	Upload Video
Tujuan	Memilih video yang akan digunakan untuk pendeteksian objek manusia
Aktor	Admin
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik tombol “Choose File”	2. Menampilkan direktori file tempat file data uji deteksi objek manusia
3. Pilih video yang akan digunakan sebagai data uji dan menekan tombol “open”	4. Menampilkan nama file data uji yang dipilih

2. *Use Case Diagram* Deteksi Objek Manusia

   Kegiatan atau interkasi antara admin dengan sistem dalam deteksi objek manusia dapat dilihat pada Tabel 3.3 berikut.

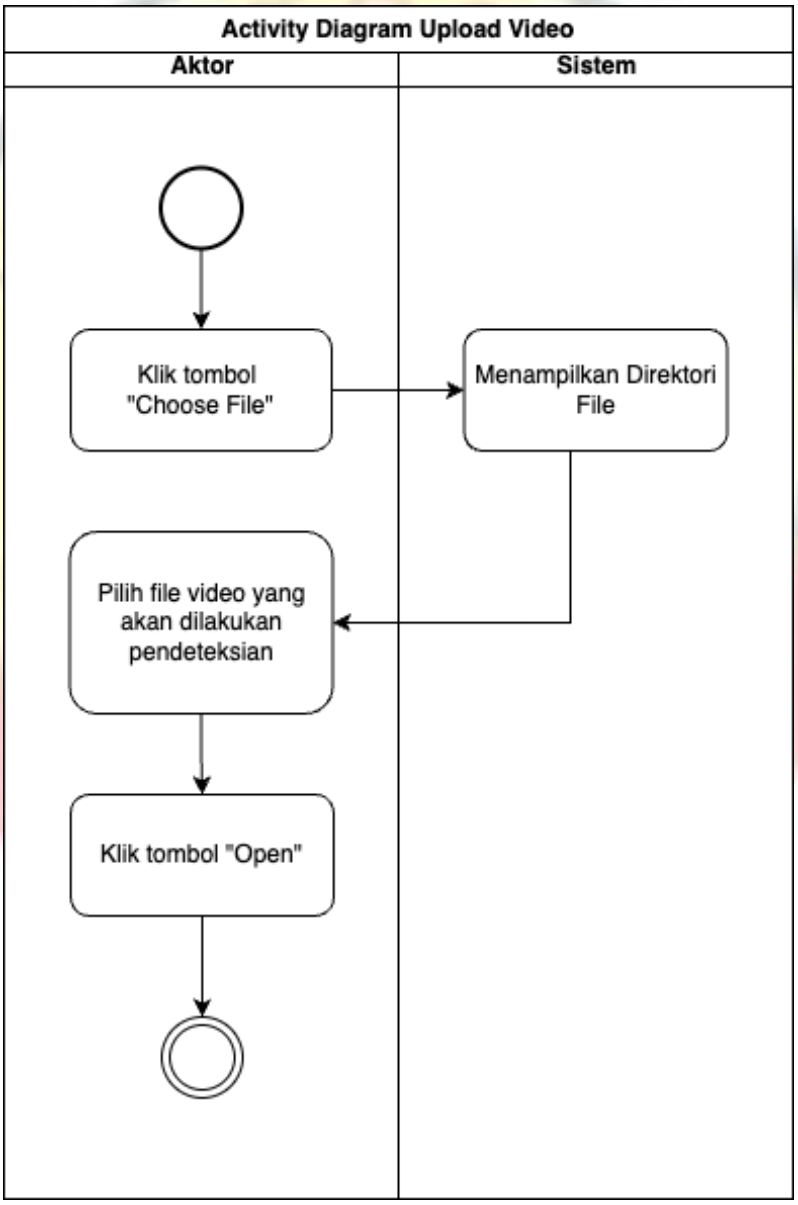
Tabel 3.3 Skenario Use Case Deteksi Objek Manusia

Identifikasi	
Nomor Use Case	02
Nama	Deteksi Objek Manusia
Tujuan	Pemroresan deteksi objek manusia pada data uji yang dipilih
Aktor	Admin
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Klik tombol “Upload”	2. Pendeteksian objek manusia pada tiap frame dari video data uji
	3. Log atau simpan tiap frame yang mengandung objek manusia ke penyimpanan sistem
	4. Menampilkan hasil deteksi dalam bentuk video

3.5.2. Activity Diagram

Activity diagram merupakan suatu proses yang menjelaskan alur kerja suatu sistem yang dibuat, bertujuan untuk menjelaskan lebih lanjut proses skenario *use case*. Berikut akan di gambarkan *activity diagram* untuk masing-masing *use case*.

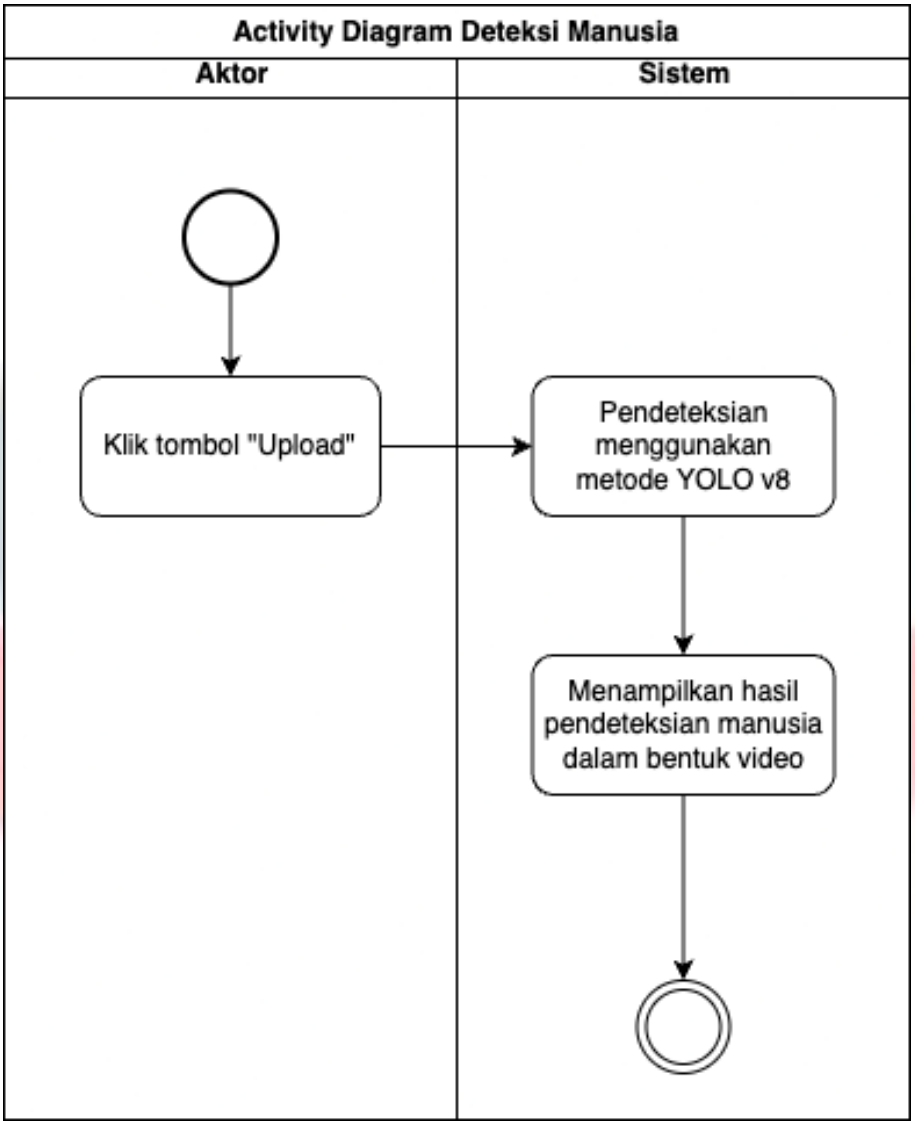
1. *Activity Diagram* Memilih Data Uji



Gambar 3.17 *Activity* Memilih Data Uji

Activity diagram pada Gambar 3.17 menjelaskan bahwa ketika aktor menekan tombol “Upload” pada halaman utama, sistem akan menampilkan halaman direktori sehingga *user* dapat memilih data uji. Proses selanjutnya pilih data uji klik tombol “Open” untuk memilih data uji tersebut dan kembali ke halaman utama.

2. *Activity Diagram* Deteksi Objek Manusia

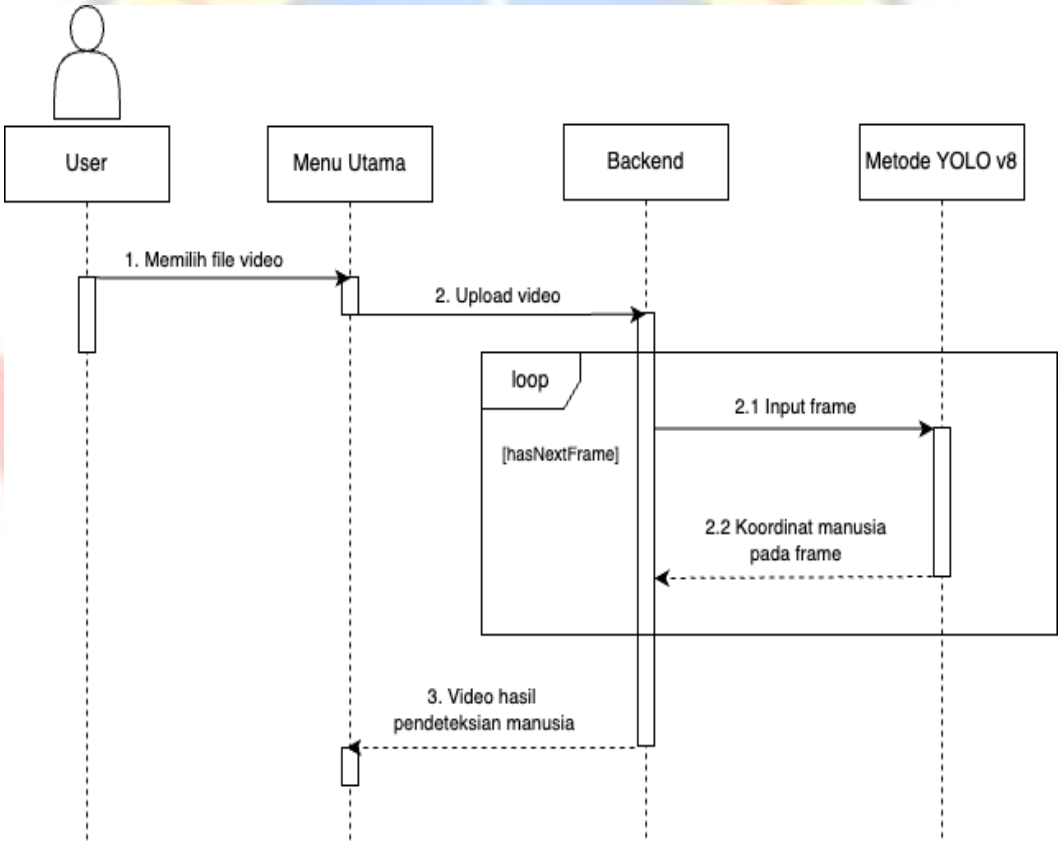


Gambar 3.18 *Activity Diagram* Deteksi Objek Manusia

Pada Gambar 3.18 menjelaskan mengenai interaksi yang terjadi antara *user* dan sistem, ketika *user* menekan tombol “Upload” pada halaman utama. Maka sistem akan melakukan proses Algoritma YOLOv8 untuk mendeteksi objek manusia.

3.5.3. Sequence Diagram

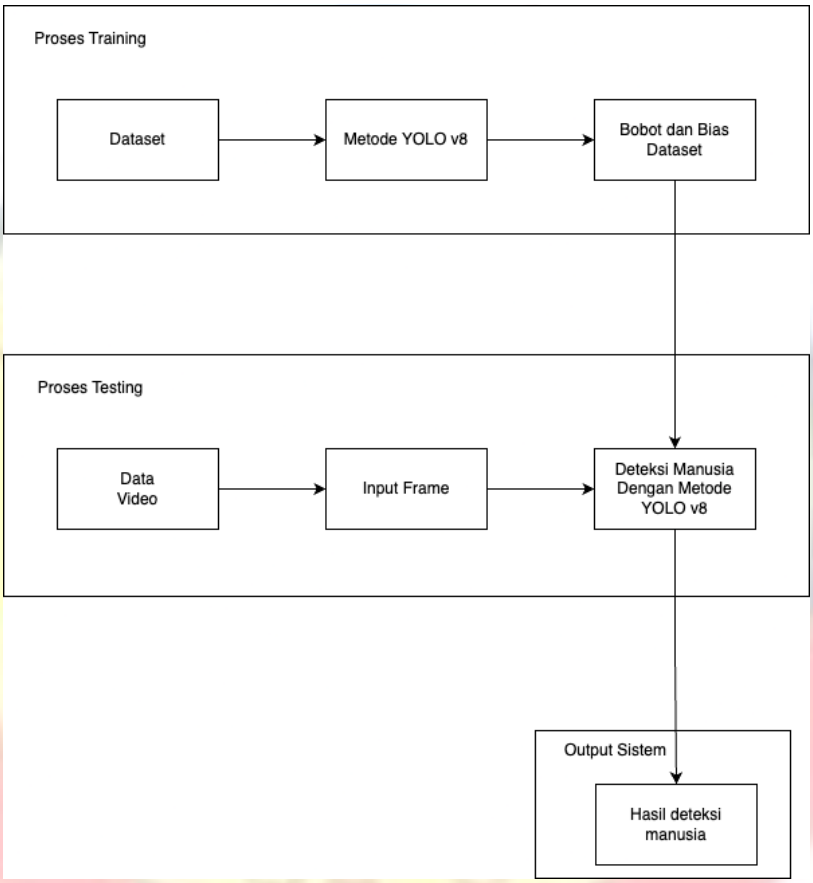
Sequence Diagram dari sistem deteksi objek manusia, user memilih data uji berupa video, kemudian data uji akan dilakukan proses Algoritma YOLOv8, Proses yang telah dilakukan menghasilkan nilai bobot untuk mendeteksi objek manusia dengan menampilkan *bounding box* dan kelas objek yang dideteksi. Gambar 3.19 merupakan sequence diagram dari sistem deteksi objek manusia.



Gambar 3.19 *Sequence Diagram* Proses Deteksi Objek Manusia

3.5.4. Blok Diagram

Dalam tahap ini dibuat blok diagram yang merepresentasikan operasi sistem dengan menunjukkan garis besar sistem yang dibuat mulai dari proses training hingga proses testing. Berikut diagram blok yang telah dibuat tersaji pada Gambar 3.20.



Gambar 3.20 Blok Diagram

Pada Gambar 3.20 ditunjukkan blok diagram untuk mendeteksi objek manusia. Berikut penjelasan dari setiap blok pada sistem deteksi objek manusia:

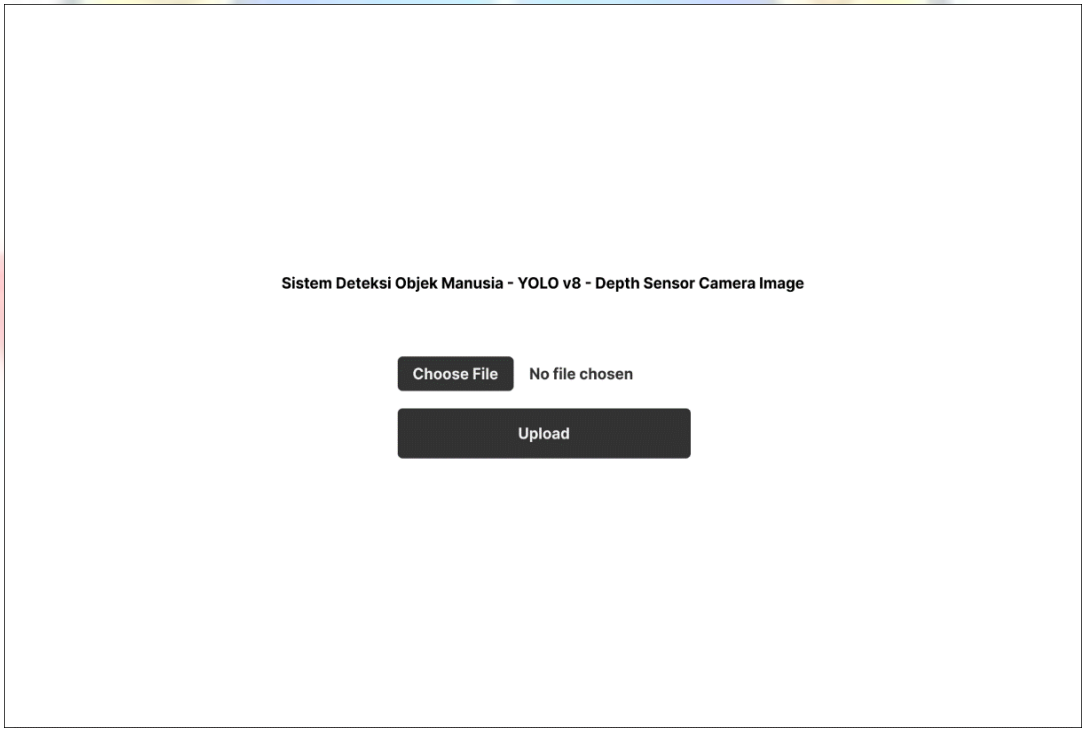
1. Pada proses training, dataset yang digunakan adalah frame dengan objek manusia.
2. Setiap masukan dataset dilakukan proses training menggunakan Algoritma YOLOv8. Dataset terdiri dari *ground truth box* yang mewakili keberadaan

objek sebenarnya pada training *image* yang ditandai dengan *bounding box* dan kelas objek.

3. Pada tahap testing, masukan data uji berupa gambar video deteksi objek manusia.
4. Masukan sistem dilakukan proses Algoritma YOLOv8 untuk menghasilkan nilai bobot dan bias pada sistem deteksi objek manusia.
5. Selanjutnya melakukan proses pencocokan nilai bobot dan bias data uji dengan dataset sehingga menghasilkan kelas objek yang dikenali.
6. Keluaran sistem yang menghasilkan deteksi objek manusia.

3.6. Pembuatan Prototype (Construction of Prototype)

Pada tahap ini dibuat perancangan antarmuka sistem deteksi objek manusia seperti yang tersaji pada Gambar 3.21.



Gambar 3.21 Antarmuka Sistem

Halaman menu pada Gambar 3.21 adalah halaman menu deteksi penggunaan manusia. Pada halaman ini *user* dapat memilih video yang akan

diproses. Proses pendeteksian pada tiap frame akan menggunakan algoritma YOLOv8 untuk mendeteksi objek manusia. Setelah selesai semua frame selesai dideteksi, sistem akan menampilkan hasil deteksi objek manusia dalam bentuk video juga.

3.7. Deployment Delivery & Feedback

Tahapan ini dilakukan untuk menguji apakah prototype yang telah dibuat sesuai dengan spesifikasi dan tujuan awal penelitian. Hal yang akan diuji pada tahap ini adalah kinerja algoritma menggunakan perhitungan precision dan recall. Detail dari pengujian akan disampaikan pada bab 4.

BAB IV
HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil dari pengembangan sistem beserta pengujiannya. Sistem dikembangkan menggunakan bahasa pemrograman *python*. Setelah pengembangan sistem selesai dilakukan, kemudian dilanjutkan oleh pengujian terhadap sistem menggunakan data uji.

4.1. Lingkungan Pengembangan

Sub-bab ini menjelaskan kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dibutuhkan untuk pengembangan sistem.

4.2.1. Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pengembangan sistem:

1. Komputer atau laptop dengan spesifikasi:
   a. Prosesor setara Apple M1 Pro
   b. RAM 16GB
2. Kamera Intel Realsense D457

4.2.2. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem deteksi objek manusia dijelaskan sebagai berikut:

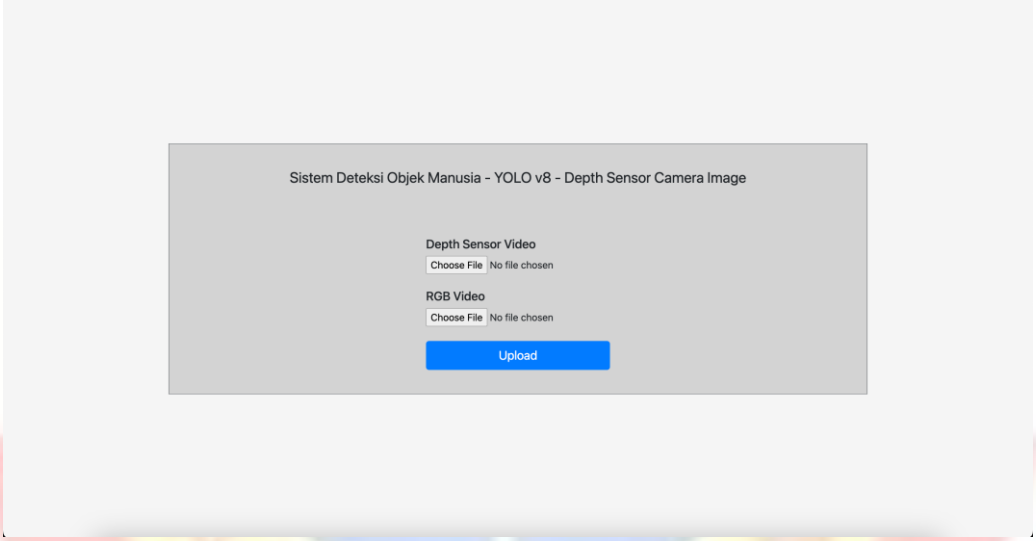
1. Mac OS Ventura 13.6 dibutuhkan sebagai sistem operasi pada laptop yang digunakan untuk menuliskan dokumentasi laporan penelitian serta menjalankan sistem deteksi objek manusia.
2. Visual Studio Code merupakan aplikasi editor untuk kode sistem deteksi objek manusia.
3. Google Colaboratory merupakan platform bahasa pemrograman python berbasis web yang digunakan untuk membuat bobot pada sistem deteksi objek manusia.

4.2. Pengembangan Sistem

Pada bagian ini, perancangan yang telah dilakukan kemudian diimplementasikan. Implementasi yang dilakukan yaitu implementasi antarmuka sistem, pemilihan data uji dan hasil deteksi.

4.2.1. Antarmuka Sistem

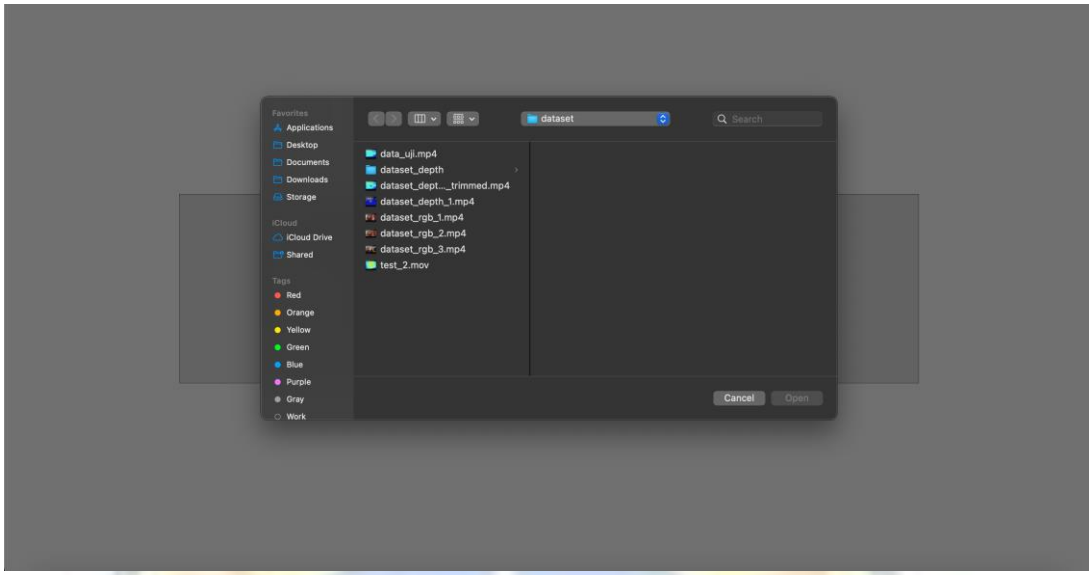
Gambar 4.1 merupakan antarmuka sistem deteksi objek manusia. Pada antarmuka ini menampilkan tombol untuk meng*upload* file uji coba pendeteksian manusia menggunakan YOLOv8.



Gambar 4.1 Antarmuka Sistem

4.2.2. Pemilihan Data Uji

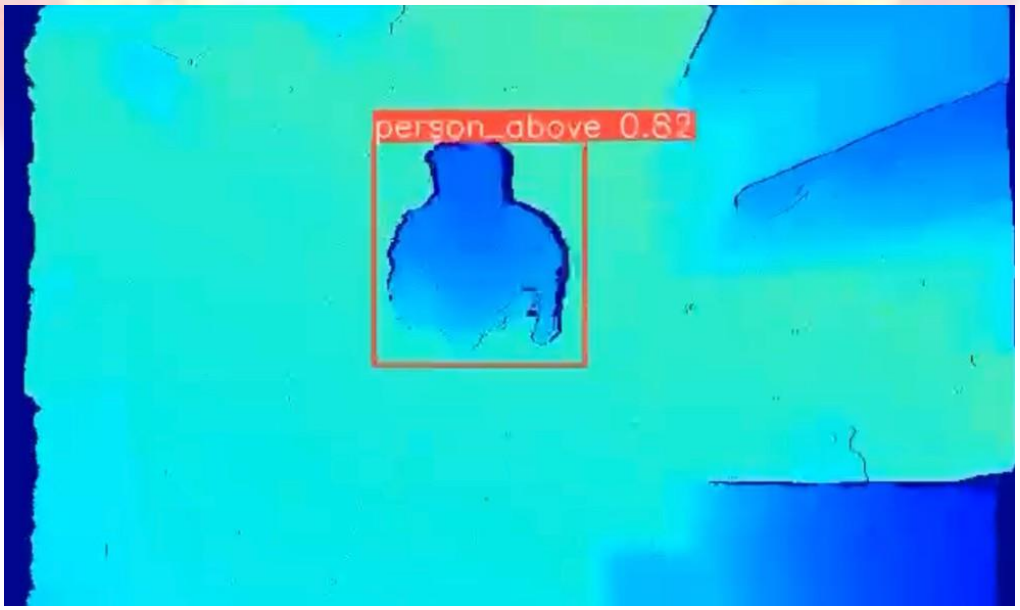
Seperti pada Gambar 4.2 antarmuka sistem terdapat tombol “Choose File” yang berfungsi untuk memilih file video pada file direktori yang akan dijadikan data uji dan tombol “Upload” yang akan mengirim file video untuk diproses oleh YOLOv8 untuk mendeteksi objek manusia pada tiap tiap frame video yang dikirim.



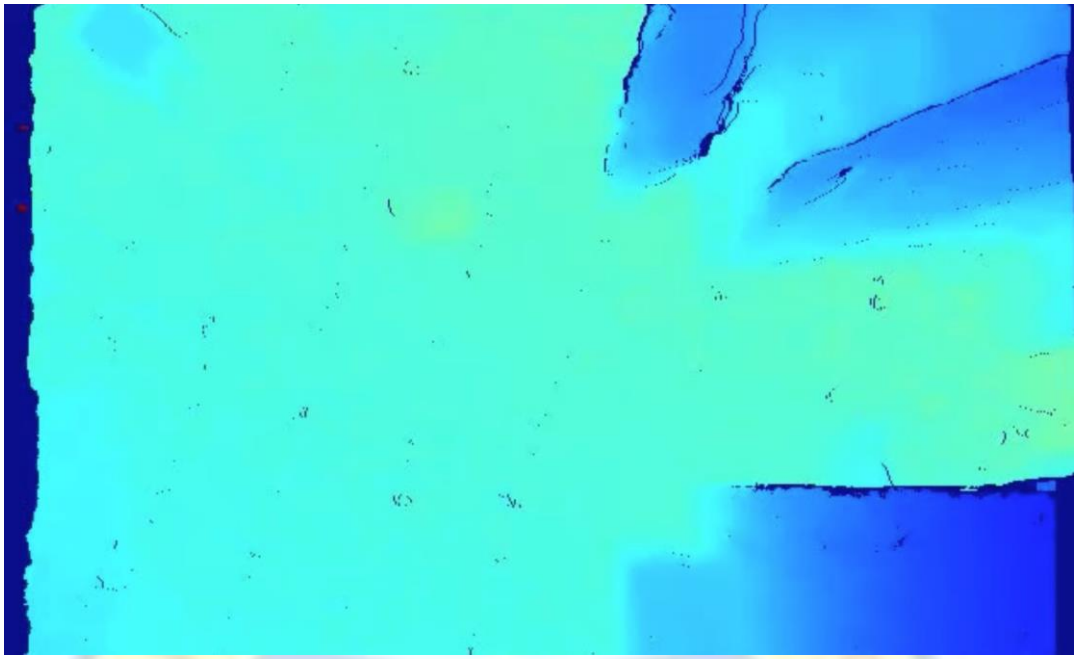
Gambar 4.2 Pemilihan Data Uji

4.2.3. Hasil Deteksi

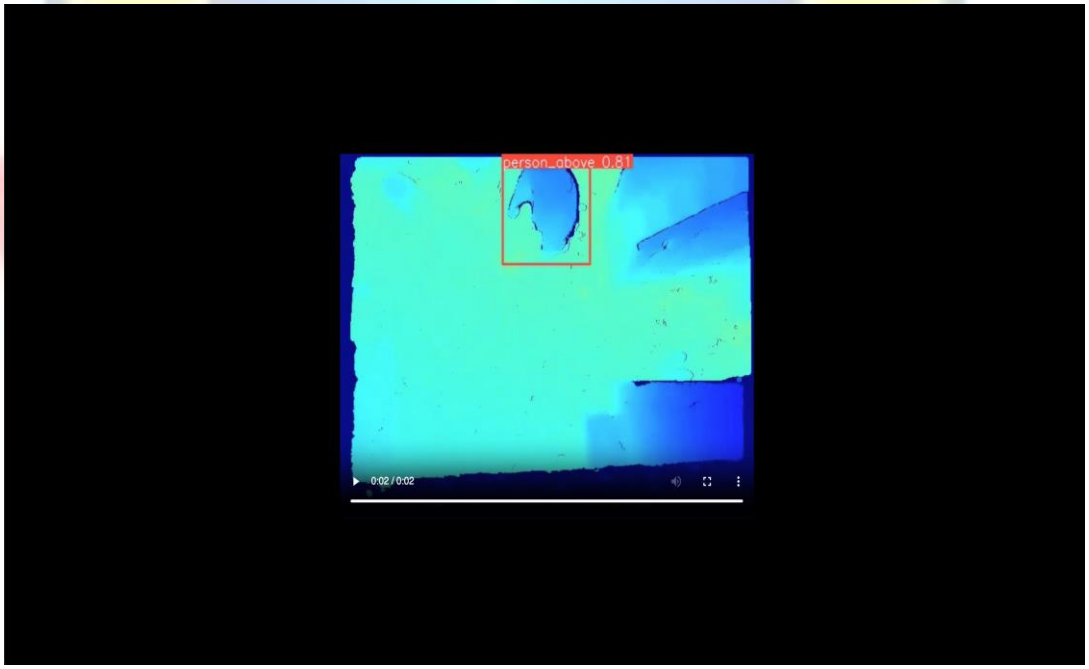
Gambar 4.3 adalah hasil deteksi objek manusia pada data uji. Hasil deteksi berbentuk *bounding box* pada objek manusia dengan label beserta nilai *confidence* untuk objek yang terdeteksi tersebut.



Gambar 4.3 Hasil Deteksi objek manusia

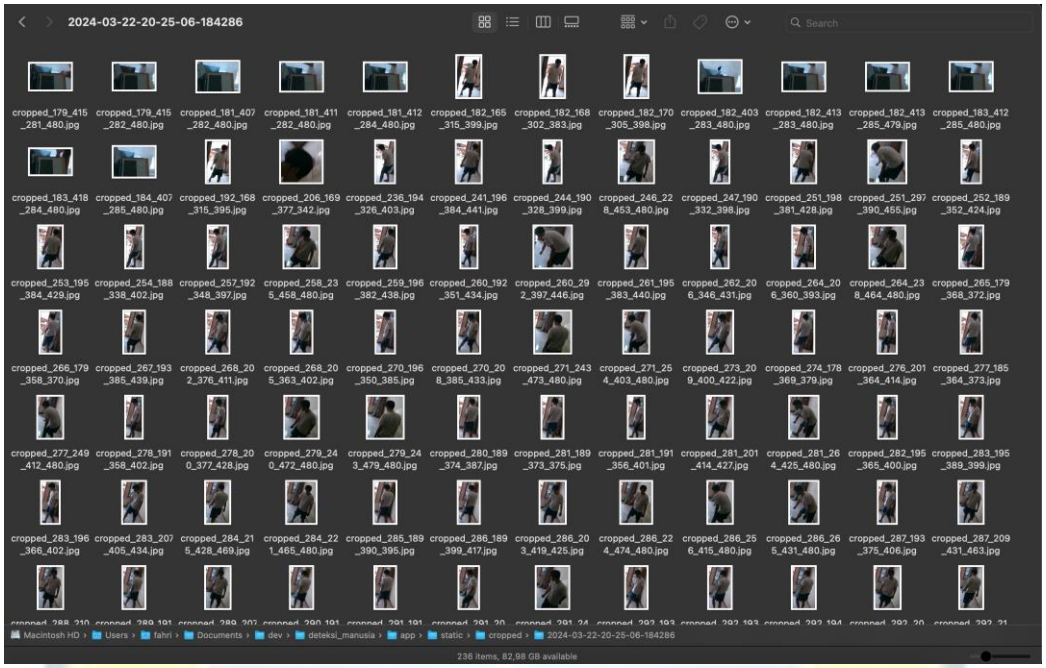


Gambar 4.4 Hasil deteksi tanpa objek manusia



Gambar 4.5 Video hasil deteksi pada antarmuka sistem

Gambar 4.4 menampilkan frame yang tidak terdeteksi objek manusia didalamnya. Gambar 4.5 menampilkan video hasil deteksi pada antarmuka sistem.



Gambar 4.6 Log objek manusia yang terdeteksi

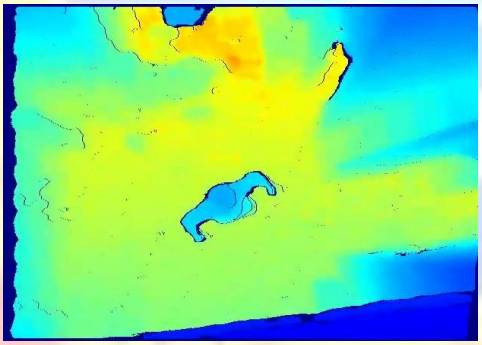
Gambar 4.6 merupakan gambar berisi objek manusia yang terdeteksi ketika proses pendeteksian. Gambar-gambar tersebut disimpan oleh sistem agar dapat digunakan dikemudian hari untuk proses penyelidikan riwayat munculnya objek manusia pada sistem.

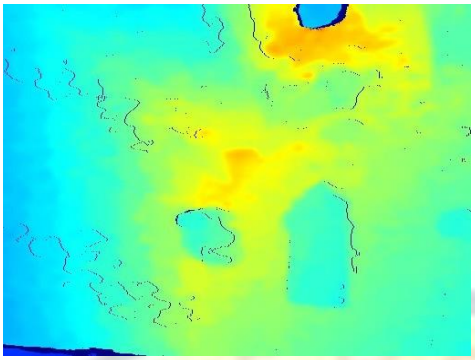
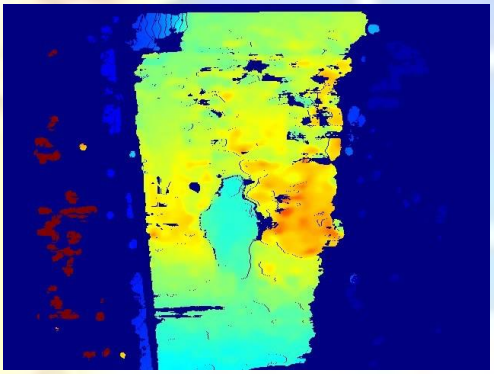
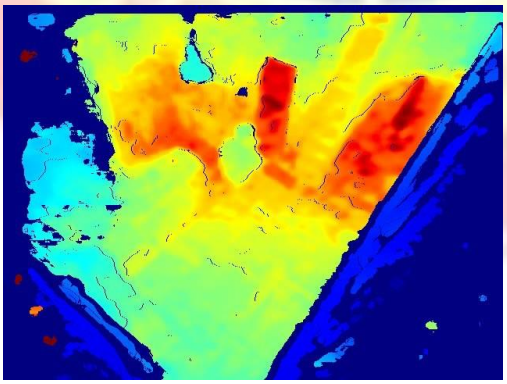
4.3. Pengujian Kinerja Algoritma

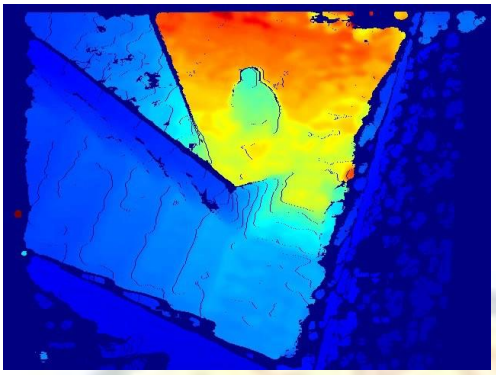
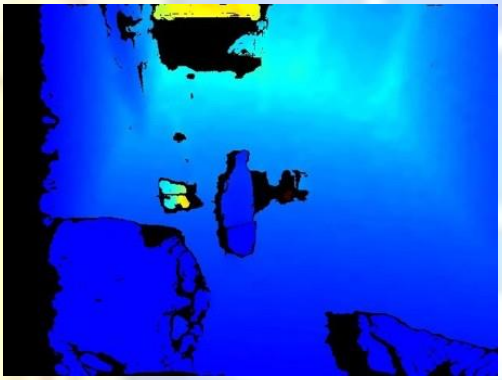
Pengujian ini bertujuan untuk mengetahui tingkat akurasi pada sistem deteksi manusia. Dalam pelaksanaan uji kinerja, dataset uji yang akan digunakan berupa lima buah video rekaman dari kamera *depth sensor*.

Groundtruth box dari dataset uji tersebut dibandingkan dengan *Predicted box* yang kemudian akan menghasilkan *Confusion matrix*. *Confusion matrix* tersebut lalu dikalkulasi untuk mendapatkan nilai *Precision*, *Recall*, *Average precision* (AP), *F-Measure*, dan *Mean Average Precision* (mAP) sesuai dengan rumus pada landasan teori. Tabel 4.1 Dataset Uji merupakan dataset untuk pengujian. Tabel 4.2 merupakan tabel hasil pengujian.

Tabel 4.1 Dataset Uji

No.	Data Input	Deskripsi
1	 Data Video 01	Video direkam dari atas dengan tinggi sekitar 2,5 meter. Total 90 frames dengan semua frame berisi objek manusia.

2	 Data Video 02	Video direkam dari dari atas dengan tinggi sekitar 2,5 meter. Total 77 frames dengan semua frame berisi objek manusia.
3	 Data Video 03	Video direkam dari tangga dengan tinggi sekitar 2 meter. Total frame 93 dengan 74 frame berisi objek manusia dan 19 frame tidak berisi objek manusia.
4	 Data Video 04	Video direkam dari kiri atas dengan tinggi sekitar 2,5 meter. Total frame 93, terdiri dari 72 frame berisi objek manusia dan 21 tidak berisi objek manusia.

5	 Data Video 05	Video direkam dari kanan atas dengan tinggi sekitar 2,5 meter. Total 75 frames terdiri dari 49 frame berisi objek manusia dan 26 tidak berisi objek manusia.
6	 Data Video 06	Video dari atas dengan ketinggian 1-2 meter. Total frames 26, dengan semua frame mengandung objek kucing untuk uji objek selain manusia, dan 11 frame berisi objek manusia.

Tabel 4.2 Hasil Pengujian

Data Input	TP	FP	FN	TN	Terdeteksi	Precision	Recall	F-Measure	Accuracy
Data Video 01	90	0	0	0	90	1,00	1,00	1,00	1,00
Data Video 02	77	0	0	0	77	1,00	1,00	1,00	1,00
Data Video 03	72	2	7	0	81	0,97	0,91	0,94	0,89
Data Video 04	72	1	0	0	73	0,99	1,00	0,99	0,99
Data Video 05	40	0	9	0	49	1,00	0,82	0,90	0,82
Data Video 06	11	2	0	0	13	0,85	1,00	0,92	0,85
Rata-rata						0,97	0,95	0,96	0,92

Tabel 4.2 Hasil Pengujian merupakan hasil perhitungan deteksi objek manusia menggunakan metode YOLOv8 pada dataset uji. Pengujian objek manusia pada kelima Data Input mendapatkan rata-rata *Precision* sebesar 97%, *Recall* 95%, *F-Measure* 96%, dan *Accuracy* 92%.

BAB V
PENUTUP

5.1. Kesimpulan

Dari penelitian yang telah dilakukan, telah dikembangkan sebuah sistem deteksi objek manusia untuk membantu meningkatkan keamanan di CV. Ateri Global Teknologi. Pengujian terhadap sistem deteksi objek manusia dengan kelima Data Input mendapatkan nilai akurasi 92% seperti pada Tabel 4.2 Hasil Pengujian.

Dari pengujian tersebut didapatkan kesimpulan sistem deteksi dapat digunakan untuk mendeteksi objek manusia. Adapun akurasi sistem tersebut dipengaruhi beberapa hal, diantaranya yaitu sudut kamera pada saat perekaman dataset dan jarak antara objek manusia dengan kamera *depth sensor*.

5.2. Saran

Berdasarkan penelitian, ada beberapa saran untuk pengembangan sistem yang dapat dilakukan sebagai berikut:

1. Menambahkan dataset dengan sudut kamera yang berbeda agar akurasi dari sistem deteksi menjadi lebih tinggi.
2. Menambahkan dataset dalam kondisi yang berbeda seperti dari luar ruangan serta pada tempat yang memiliki banyak objek lain untuk menambah akurasi pada kondisi yang berbeda.

DAFTAR PUSTAKA

Kristanto, A., 2008. *Perancangan Sistem Informasi dan Aplikasinya.* Yogyakarta: Gava Media.

Rangkuti, M., 2023. *Artikel dan Berita.* [Online]
Available at: https://umsu.ac.id/artikel/mengenal-artificial-intelligence-ai-pengertian-sejarah-kegunaan-dan-contoh-penerapannya
[Diakses 21 Januari 2024].

Sutabri, T., 2012. *Analisis sistem informasi.* Yogyakarta: s.n.

Budiarjo, D. D., 2020. IMPLEMENTASI SISTEM CERDAS PADA OTOMATISASI PENDETEKSIAN JENIS KENDARAAN DI JALAN RAYA. *Repository USM.*

Rusydi Umar, Imam Riadi, Purwono, 2020. Klasifikasi Kinerja Programmer pada Aktivitas Media Sosial dengan Metode Support Vector Machines. *CYBERNETICS,* Volume 4.

Yanto, Y., Aziz, F. & Irmawati, I., 2023. YOLO-V8 PENINGKATAN ALGORITMA UNTUK DETEKSI PEMAKAIAN MASKER WAJAH. *JATI - Jurnal Mahasiswa Teknik Informatika.*

RAHMAN, A. A., AGUSTIN, S. D., IBRAHIM, N. & KUMALASARI, N. C., 2022. Perbandingan Algoritma YOLOv4 dan Scaled YOLOv4 untuk Deteksi Objek pada Citra Termal. *MIND JOURNAL.*

Zhou, C. et al., 2022. Human Position Detection Based on Depth Camera Image Information in Mechanical Safety. *Hindawi.*

Putro, E. C., Awangga, R. M. & Andarsyah, R., 2000. *Tutorial Object Detection People With Faster region-Based Convolutional Neural Network(Faster R-CNN).* s.l.:s.n.

Openg, J. B. J. R., Hiswati, M. E. & Hamzah, H., 2022. Klasifikasi Unggas Ordo Anseriformes Berdasarkan Citra Menggunakan Metode Deep Learning Dengan Algoritma Convolutional Neural Network (CNN). *SinTaKS.*

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A., 2015. You Only Look Once: Unified, Real-Time Object Detection. *arxiv.*

Drantantiyas, N. D. G. et al., 2023. Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8. *JASIEK,* 5(2).

Motwani, N. P. & S, S., 2023. Human Activities Detection using DeepLearning Technique- YOLOv8. *ITM Web Conf.*.

Nugroho, P. A., Fenriana, I. & Arijanto, R., 2020. IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL (CNN) PADA EXPRESI MANUSIA. *ALGOR.*

Bai, R., Shen, F., Wang, M. & Lu, J., 2023. Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity. *ResearchGate.*

Santoso, A. & Ariyanto, G., 2018. Implementasi Deep Learning berbasis Keras untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro,* Volume 18.

N.Nufus, et al., 2021. Sistem Pendeteksi Pejalan Kaki Di Lingkungan terbatas berbasis SSD MobileNet V2 Dengan Menggunakan Gambar 360 Ternormalisasi. *Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia.*

Ilahiyah, S. & Nilogiri, A., 2018. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO.*

Ultralytics, 2024. *Ultralytics YOLOv8 Docs.* [Online]
Available at: https://docs.ultralytics.com/reference/data/augment/
[Diakses February 2024].

Kani, 2020. *Pengantar Algoritma dan Pemrograman.* Edisi Kesatu penyunt. Banten: Universitas Terbuka.

geeksforgeeks, 2024. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/introduction-convolution-neural-network/
[Diakses Februari 2024].

Liang, G. et al., 2021. Malicious Packages Lurking in User-Friendly Python Package Index. *IEEE.*

Suradi, A. A. M. et al., 2023. Sistem Deteksi Kantuk Pengemudi Mobil Berdasarkan Analisis Rasio Mata Menggunakan Computer Vision. *JUKI : Jurnal Komputer dan Informatika,* Volume 5.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach.* s.l.:s.n.

Sidharta, H. A., 2017. *Binus University.* [Online]
Available at: https://binus.ac.id/malang/2017/10/introduction-to-open-cv/
[Diakses Februari 2024].

RevoU, 2024. [Online]
Available at: https://revou.co/kosakata/supervised-learning
[Diakses 2024].

Dharwiyanti, S. & Wahono, R. S., 2023. *Pengantar Unified Modeling Language (UML).* s.l.:s.n.

Bergeron, B., 2003. *Essentials Of Knowledge Management.* s.l.:s.n.

Kumar, V. & Chadha, A., 2012. Mining Association Rules in Student's Assessment Data. *International Journal of Computer Science.*

TADIC, V. et al., 2019. Application of Intel RealSense Cameras for Depth Image Generation in Robotics. *WSEAS TRANSACTIONS on COMPUTERS,* Volume 18.

Bisong, E., 2019. Getting Started with Google Cloud Platform. Dalam: *Building Machine Learning and Deep Learning Models on Google Cloud Platform.* 2019: Apress, pp. 59-64.

Setiyani, L., 2021. Desain Sistem : Use Case Diagram. *Seminar Nasional : Inovasi & Adopsi Teknologi.*

LAMPIRAN

Remove gambar tanpa pelabelan/objek dari dataset

```
import os

TXT_ROOT = os.path.join('valid', 'labels')
IMG_ROOT = os.path.join('valid', 'images')

all_txt_files = os.listdir(TXT_ROOT)

zero_counter = 0
txt_without_obj = []
for i, file_name in enumerate(all_txt_files):
    file_path = os.path.join(TXT_ROOT, file_name)
    with open(file_path, 'r') as f:
        lines = f.readlines()
        if len(lines) == 0:
            zero_counter += 1
            txt_without_obj.append(file_name.split('.txt')[0])
    f.close()

all_img_files = os.listdir(IMG_ROOT)
zero_img_counter = 0
for i, file_name in enumerate(all_img_files):
    file_path = os.path.join(IMG_ROOT, file_name)
    if '.'.join(file_name.split('.')[:-1]) in txt_without_obj:
        zero_img_counter += 1
        os.remove(os.path.join(IMG_ROOT, file_name))
        os.remove(os.path.join(TXT_ROOT, '.'.join(file_name.split('.')[:-1])+'.txt'))
```

Remove duplikat file Bernama sama pada dataset

```
import os
import argparse

def buildArgParser():
    parser = argparse.ArgumentParser()
    parser.add_argument('-pa', '--folder-path-a', type=str,
                        required=True,
                        help='Path to folder A')
    parser.add_argument('-pb', '--folder-path-b', type=str,
                        required=True,
                        help='Path to folder B')

    return parser

def remove_duplicates(folder_path_a, folder_path_b):
    files_a = os.listdir(folder_path_a)
    files_b = os.listdir(folder_path_b)

    duplicates = set(files_a) & set(files_b)

    for duplicate_file in duplicates:
        file_path_a = os.path.join(folder_path_a, duplicate_file)
        file_path_b = os.path.join(folder_path_b, duplicate_file)

        if os.path.isfile(file_path_a) and os.path.isfile(file_path_b):
            os.remove(file_path_b)

def main():
    args = buildArgParser().parse_args()
    remove_duplicates(args.folder_path_a, args.folder_path_b)

if __name__ == "__main__":
    exit(main() or 0)
```

Remove gambar berakhiran genap dari dataset

```python
def remove_odd_images(folder_path):
    # List all files in the folder
    files = os.listdir(folder_path)

    # Loop through each file
    for file in files:
        if "video" not in file:
            continue

        # Split file name to extract the number part
        file_number = file.split('_')[-1]
        # Extract the number and convert it to an integer
        number = int(file_number.split('.')[0])

        # Check if the number is odd
        if number % 2 =/= 0:
            # Create the full file path
            file_path = os.path.join(folder_path, file)
            # Remove the file
            os.remove(file_path)
```

Remove setengah dataset

```python
def reduce_by_half(folder_path):
    files = os.listdir(folder_path)
    for (index, file) in enumerate(files):

        if index % 2 == 0:
            # Create the full file path
            file_path = os.path.join(folder_path, file)
            # Remove the file
            os.remove(file_path)
```

Halaman Utama Antarmuka Sistem

```
<body class="align-content-center">
  <div class="container align-content-center p-4 border border-dark" style="background-color: lightgray;">
    <h1 class="h4 m-3 font-weight-normal d-flex justify-content-center" style="height: 80px;">
      Sistem Deteksi Objek Manusia - YOLO v8 - Depth Sensor Camera Image
    </h1>

    <form class="form-signin" method=post enctype=multipart/form-data>
        <input type="file" name="file" class="form-control-file" id="inputfile">
        <br/>
        <button class="btn btn-lg btn-primary btn-block" type="submit">Upload</button>
    </form>
  </div>
```

Validasi input video uji

```
<script type="text/javascript">
  $('#inputfile').bind('change', function() {
      let ext = $('#inputfile').val().split('.').pop().toLowerCase();
      if($.inArray(ext, ['mp4']) == -1 && $.inArray(ext, ['avi']) == -1) {
        $("#inputfile").val(null);
        alert('video format not allowed!');
      }
  });
</script>
```

Ekstrak video menjadi *frames*

```
def video_to_frames(video_path):
    print(get_time(), "converting video to frames")
    frames = []
    vidcap = cv2.VideoCapture(video_path)
    success, image = vidcap.read()

    while success:
        frames.append(image)
        success, image = vidcap.read()

    print(get_time(), "done converting. frames: {}".format(len(frames)))
    return frames
```

Konversi hasil prediksi menjadi video

```python
def images_to_video(folder_path):
    print(get_time(), "converting images to video")
    now_time = datetime.datetime.now().strftime(DATETIME_FORMAT)
    os.makedirs('./static/video', exist_ok=True)
    video_name =  now_time + '.mp4'
    video_path = os.path.join('./static/video', video_name)

    images = [img for img in os.listdir(folder_path) if img.endswith(".jpg")]
    frame = cv2.imread(os.path.join(folder_path, images[0]))
    height, width, layers = frame.shape

    video = cv2.VideoWriter(video_path, cv2.VideoWriter_fourcc(*'avc1'), 5, (width, height))

    for image in images:
        video.write(cv2.imread(os.path.join(folder_path, image)))

    video.release()

    return video_path
```

Proses pendeteksian

```python
if request.method == "POST":
    if "file" not in request.files:
        return redirect(request.url)
    file = request.files["file"]
    if not file:
        return
  Robin Cole, 15 months ago • Merge pull request #39 from robmarkcole/ISS
    filename = file.filename
    file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
    file.save(file_path)

    frames = video_to_frames(file_path)
    print(get_time(), "start prediction")
    result_folder = './static/results/{}'.format(get_time())
    model.predict(
        source=frames,
        imgsz=640,
        save=True,
        project=result_folder
    )

    video_path = images_to_video(os.path.join(result_folder, 'predict'))
    return redirect(video_path)
```

KARTU BIMBINGAN SKRIPSI
S1 – TEKNIK INFORMATIKA
UNIVERSITAS SANGGA BUANA - YPKP

TAHUN AJAR	Ganjil 2023/2024
NPM	2113191049
NAMA	Fahri Muhamad Zulkarnaen
PEMBIMBING	Bambang Sugiarto, ST., M.T.
JUDUL	SISTEM DETEKSI MANUSIA MENGGUNAKAN ALGORITMA YOLO V8 DAN DEPTH SENSOR CAMERA (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)

PHOTO 3X4

NO	TANGGAL	POKOK BAHASAN	PARAF PEMBIMBING
1	29/11/23	Bab I	
2	7/12/23	Bab II	
3	14/12/23	Bab III	
4	16/02/24	Bab IV	
5	23/02/24	Revisi bab IV	
6	26/02/24	Bab V	
7	01/03/24	Revisi	
8	02/03/24	Revisi	
9	02/03/24	Revisi	
10	02/03/24	Diskusi Presentasi	

Cat :

1. Minimal bimbingan sebanyak 8x.
2. Kartu ini dikumpulkan sebagai syarat sidang beserta berkas yang lainnya.

Bandung, ...20...

Pembimbing

Sugianto

(Bambang Sugiarto, ST., M.T.)

UPT PERPUSTAKAAN
UNIVERSITAS SANGGA BUANA YPKP
Jl. PHH Mustofa No. 68 Bandung Gedung E Lantai 5
Email: library@usbypkp.ac.id Website: perpustakaan.usbypkp.ac.id

Surat Keterangan Cek Plagiarisme

Nomor : 103/III/SKCP/USB-YPKP/2024

Sehubungan dengan kewajiban **Cek Plagiarisme** dengan ***similarity check maximal 25%*** sebagai salah satu kelengkapan persyaratan administrasi bagi mahasiswa tingkat akhir, dengan ini UPT Perpustakaan Universitas Sangga Buana menerangkan bahwa:

Nama	: FAHRI MUHAMAD ZULKARNAEN
NPM	: 2113191049
Program Studi	: S1 Teknik Informatika
Judul Karya Tulis Ilmiah	:"SISTEM DETEKSI OBJEK MANUSIA MENGGUNAKAN ALGORITMA YOLO V8 BERBASIS DEPTH SENSOR CAMERA (STUDI KASUS: CV. ATERI GLOBAL TEKNOLOGI)"
Tanggal Cek Turnitin	: 02-Mar-24
Status	: Lulus dengan **19%** *Similiraty Check*

Adalah benar telah dilakukan ***similarity check*** sebagaimana data tersebut diatas, dan surat ini dibuat berdasarkan keadaan yang sebenar benarnya, untuk bisa dipergunakan sebagaimana mestinya.

Bandung, 02-Mar-24

Kepala UPT Perpustakaan

Widyapuri Prasastiningtyas, S.Sos., M.I.kom.

NIP. 432.200.173