

BAB II TINJAUAN PUSTAKA

2.1. Sistem *Real-Time*

Suatu sistem yang dapat mengendalikan sistem fisik dikenal sebagai sistem *real-time* [16]. Sistem *real-time* merupakan bentuk khusus dari sistem *online*. Sebuah sistem dinyatakan *real-time* jika tidak selalu mengedepankan ketepatan suatu proses, tetapi juga dengan jeda atau jarak waktu proses tersebut dilakukan. Sehingga suatu sistem *real-time* ialah sistem yang menggunakan sistem *deadline*, ialah pekerjaan yang semestinya beres dengan periode waktu yang diberikan. Pada sistem *real-time*, digunakan batasan waktu. Sistem dinyatakan gagal jika melewati batasan yang ada. Sistem *real-time* banyak diterapkan dalam berbagai macam kebutuhan kita sehari-hari contohnya sistem tersebut dimasukan di dalam alat khusus seperti di kamera telepon genggam, pemutar musik, ataupun di mobil dan pesawat [16].

Terdapat dua bentuk sistem *real-time*, yaitu sistem *hard real-time* dan sistem *soft real-time*. Pada *hard real-time*, sistem menanggung tugas yang berat akan diselesaikan secara cepat. Dalam sistem ini penyimpan kedua terbatas atau tidak terpakai, data langsung dikirim ke *memory* atau *read-only memory* (ROM) dalam waktu yang cepat. Dalam *hard real-time* sistem terjadi masalah pada sistem pembagian waktu dan tidak ditunjang oleh sistem operasi tujuan umum. Gambaran lainnya ialah *soft real-time* dimana *task* yang berat memperoleh pengetumaan yang lebih tinggi dibandingkan dengan tugas lain dan setelah tugas selesai maka tugas yang diutamakan ini akan diselesaikan. Sistem ini tidak banyak dijumpai pada bidang industri robotik. Sistem *real-time* sangat bermanfaat pada aplikasi komunikasi dan realita *virtual* [16].

Suatu *system* dikatakan *real-time* jika *system* tersebut dapat mencakup pengerjaan program atau aplikasi dengan waktu yang telah ditetapkan. Kelebihan suatu sistem *real-time*, yaitu:

- a. Mempunyai ketentuan waktu dan mencukupi tenggang waktu.
- b. Memiliki waktu yang bisa diperkirakan.

- c. Terfokus pada hal yang utama saja, sehingga dapat menemukan tingkat efisiensi waktu.

Selain kelebihan, adapun beberapa kekurangan *real-time system*, yaitu:

- a. Mempunyai hanya satu ara tujuan, contohnya memindahkan satu lagu dari perangkat laptop ke pemutar musik.
- b. Biasanya sistem mempunyai ruang fisik yang terbatas.
- c. Metode ini harus memenuhi persyaratan saat yang ditentukan dengan menggunakan algoritma.

2.2. Bahasa Pemrograman Python

Python diciptakan oleh Guido Van Rossum pertama kali di *Scitching Mathematisch Centrum* (CWI) di Belanda pada awal tahun 1990-an. Bahasa Python termotivasi dari bahasa pemrograman ABC. Sampai saat ini, Guido masih menjadi pembuat utama untuk Python, walaupun *open source* dan bisa semua orang juga berpartisipasi dalam pengembangannya. Pada tahun 1995, Guido meneruskan pengerjaan Python di *Corporation for National Research Initiative* (CNRI) di Virginia Amerika.

Pada bulan mei tahun 2000, Guido beserta rekannya pindah ke *BeOpen.com* dan membuat tim *BeOpen PythonLabs*. Pada bulan oktober tahun 2000, regu Python pindah ke kreasi digital dan lebih dikenal menjadi Perusahaan Zope. Di tahun 2001, dibuatlah sebuah kelompok Python yaitu *Python Software Foundation* (PSF). PSF adalah sekelompok nirlaba yang rancang istimewa agar bisa dalam semua hal yang bersangkutan dengan hak intelektual Python. Perusahaan Zope menjadi anggota pendukung dari PSF [17].

Python adalah bahasa pemrograman yang mempunyai pandangan mempunyai banyak kegunaan dengan filsafat proses yang bertujuan utama pada pemahaman kode secara cepat. Python diakui sebagai bahasa yang menyatukan keahlian dengan sistem kode yang jelas, Python pun di sponsori oleh organisasi yang besar.

Python bisa mempunyai banyak pembuatan rangka dari awal pemrograman, khususnya tetapi tidak terbatas pada pemrograman berfokus objek, bahasa yang mempunyai perintah, dan bahaya yang fungsional. Beberapa fitur yang ada pada Python adalah salah satu bahasa pemrograman dinamis yang dilengkapi pengaturan

penyimpanan otomatis. Sama kasusnya pada bahasa pemrograman dinamis lainnya, Python kebanyakan dipakai layaknya bahasa *script* walaupun pada kenyataannya pemakaian bahasa ini lebih luas pembahasan makna manfaat yang tidak dilakukan dengan menggunakan bahasa *script*. Python bisa berfungsi untuk beberapa kepentingan pengembangan *software* dan dapat berjalan di berbagai penyedia program operasi sistem [18]. Python memiliki keunggulan dibandingkan bahasa pemrograman lainnya, berikut adalah keunggulannya:

1. Python mempunyai gagasan bentuk yang baik dan mudah digunakan, kode Python dibuat lebih gampang untuk dipahami, dijalankan, dipakai secara berulang, dan dijaga. Di luar dari itu, Python lebih mendorong pemrograman bertemakan suatu objek dan pemrograman bisa berfungsi sebagai apapun.
2. Python mampu menambah kemampuan dan ketepatan waktu untuk penggunaannya, untuk mendapatkan hasil kode yang sesuai, kode Python mempunyai lebih sedikit kode yang dibuat dibandingkan memakai pemrograman bahasa yang lain misalnya Java maupun *C*, *C++*, dan *C#*.
3. Program yang dibuat melalui Python dapat diterapkan di kebanyakan sistem operasi seperti Unix, Windows, Mac *OS X*, dan juga perangkat komunikasi lainnya.
4. Python mempunyai banyak bantuan pustaka yang lebih di analisis oleh pihak lain, contohnya rujukan untuk pengolahan situs, pemutakhiran aplikasi visual berintegrasikan GUI, penelitian lebih lanjut dalam suatu permainan, dan lain-lain.
5. Menggunakan intruksi tertentu, kode Python bisa digabungkan dengan aplikasi yang dibuat di bahasa pemrograman lain. Seperti halnya kode Python bisa dijalankan dari kode *C/C++*, dan tak lepas dari perkembangan *.NET framework*.
6. Python dapat digunakan secara gratis dan *open source*, walaupun dipakai untuk kebutuhan pemasaran [19].

Terdapat hal penting dalam Python yaitu penyunting naskah. Penyunting naskah ialah perangkat lunak program komputer yang menjadikan pemakainya untuk menciptakan, menyunting berkas *text* dalam rangka susunan teks. Penyunting teks dapat dipakai untuk menciptakan aplikasi dan menyunting sumber kode dari bahasa pemrograman. Penyunting teks dapat digunakan dalam menyusun halaman situs dan dapat menciptakan program. Kegunaan penyunting naskah antara lain:

1. Menemukan kalimat atau bahasa dalam suatu berkas.

2. Membuat, mengeksekusi, dan untuk mencari serta mengurangi kesalahan dalam suatu aplikasi.
3. Mengelompokkan data yang yang tersedia.
4. Memadukan berkas satu dan yang lain.

Python memiliki penyunting naskahnya nya sendiri yaitu IDLE. *Integrated DeveLopment Environment* (IDLE) adalah penyunting yang tersedia di Python yang dapat di pakai untuk membuat Python kode. IDLE mempunyai mode *shell* atau interaksi mode yang dapat dipakai untuk melakukan suatu tugas dengan hasil output yang bisa langsung melihat hasilnya. IDLE pun ialah asli diciptakan oleh Guido van Rossum dari bahasa Python. IDLE mempunyai kelebihan yang sangat bermanfaat bagi penggunanya, diantaranya:

1. *Editor multi-window* dengan fitur *syntax highlighting*, *autocomplete*, *smart indent*, dan lain–lain.
2. *Shell* interaksi dengan kelebihanya *syntax highlighting*.
3. Mempunyai kelebihan *debugger* agar bisa *debug listing* program [20].

2.3. Image Processing

Image processing atau pengolahan citra merupakan mekanisme pengolahan gambar yang merubah gambar *input* sebagai gambar yang berbeda lain supaya keluarannya mempunyai mutu yang lebih bagus dibandin mutu gambar *input*. Hal tersebut dapat berguna sebagai syarat menaikkan mutu sebuah gambar, menghapuskan kerusakan terhadap gambar, mengetahui objek, serta menjadikan satu oleh kelompok gambar lainnya. Saat ini, *image processing* adalah satu dari banyak pengetahuan metode ilmiah yang bertambah cepat dan tidak bisa dipisahkan oleh bagian visi komputer [21].

Kualitas gambar masukan merupakan peran penting dalam keberhasilan komputasi gambar, karena semakin tinggi kualitas gambarnya, maka lebih mudah komputasi dijalankan. Jadi untuk meningkatkan kualitas gambar masukan, metode komputasi yang tepat digunakan ialah *image processing*. Langkah-langkah yang diperlukan seperti menghilangkan *noise*, mengoreksi pada geometris, meningkatkan tepi dan kontras, serta mengoreksi iluminasinya [21].

2.4 OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan *machine learning* dan bersifat bebas sumber. OpenCV versi pertama diluncurkan pada tahun 1999, asal mula nya mementingkan perpustakaan dari Intel *Image Processing Library* yang lalu ketergantungannya sebagaimana dilenyapkan agar terbentuklah OpenCV layaknya seperti perpustakaan yang berdiri sendiri. OpenCV menunjang banyak program, bisa menunjang seperti Linux ataupun Windows, pada saat ini sudah bisa menunjang Android dan MacOSX. *Library* OpenCV amat banyak digunakan untuk pemrosesan citra karena mempunyai sangat banyak algoritma, yaitu sebesar 2500 algoritma, dimana algoritma itu dapat di pakai sebagai mengolah citra mulai dari deteksi wajah, pengenalan wajah, *object tracking*, deteksi tepi hingga *Kalman filtering*. OpenCV juga merupakan *library open source* yang digunakan untuk bahasa C++, Python, Java, dan Matlab. OpenCV mempunyai puluhan ribu orang sebagai dari sekelompok pengguna yang dapat diperkirakan lebih dari belasan juta kali diunduh [22]. OpenCV memiliki kelebihan pada *library* OpenCV diantaranya:

1. Membuat gambar data (tempat penyimpanan, menghilangkan penyimpanan, menyalin citra, pengaturan dan merubah citra).
2. Gambar/Vidio.
3. Sampling gambar dan transformasi.
4. Metode untuk AI dan *machine learning*.
5. Membuat matrikulasi dan besaran dan ada pada *routines linear algebra* (produk, pemecah masalah, nilai eigen, SVD).
6. *Basic* pemrosesan gambar seperti penyaringan, deteksi tepi, pengenalan sisi, proses seleksi hingga pengalihan, perubahan warna, membuat struktural, grafik statistik dan gambar piramida.
7. Analisa struktur.
8. Mengatur potret.
9. Mengetahui gerakan.
10. Mengidentifikasi hal.
11. Dasar antarmuka pengguna seperti citra, rekaman, papan ketik dan lainnya.
12. Penandaan gambar seperti garis, kerucut, polygon, dan gambar teks.

2.5. Deteksi Objek

Ketika manusia melihat suatu objek, kemampuan kita untuk dengan cepat mengenali gambar dalam citra, menentukan posisi objek tersebut, dan memahami situasi yang sedang berlangsung adalah salah satu aspek yang sangat kuat dari kemampuan visual kita. Basis visual yang dimiliki oleh orang yang terampil dan tepat dalam pengamatan ini memungkinkan kita untuk menjalankan tugas yang kompleks, bahkan dalam situasi yang sulit sekalipun. [23]. Dengan adanya algoritma deteksi objek yang canggih, komputer dapat digunakan dalam berbagai aplikasi, termasuk pengawasan keamanan, kendaraan otonom, pengolahan medis, pengenalan wajah, dan banyak lagi. Kemampuan ini dapat meningkatkan efisiensi dan akurasi dalam berbagai tugas, memungkinkan komputer untuk bekerja dengan citra dan video dalam konteks yang lebih mendalam dan bermakna.

Proses ini sering melibatkan teknik-teknik seperti ekstraksi fitur, pemodelan statistik, penggunaan algoritma pembelajaran mesin, atau jaringan saraf tiruan dalam kasus algoritma deteksi objek yang telah dibahas sebelumnya. Tujuan utamanya adalah untuk mengidentifikasi dan menentukan lokasi serta karakteristik objek dengan akurasi tinggi dalam citra atau video.

2.6. Tensorflow Lite

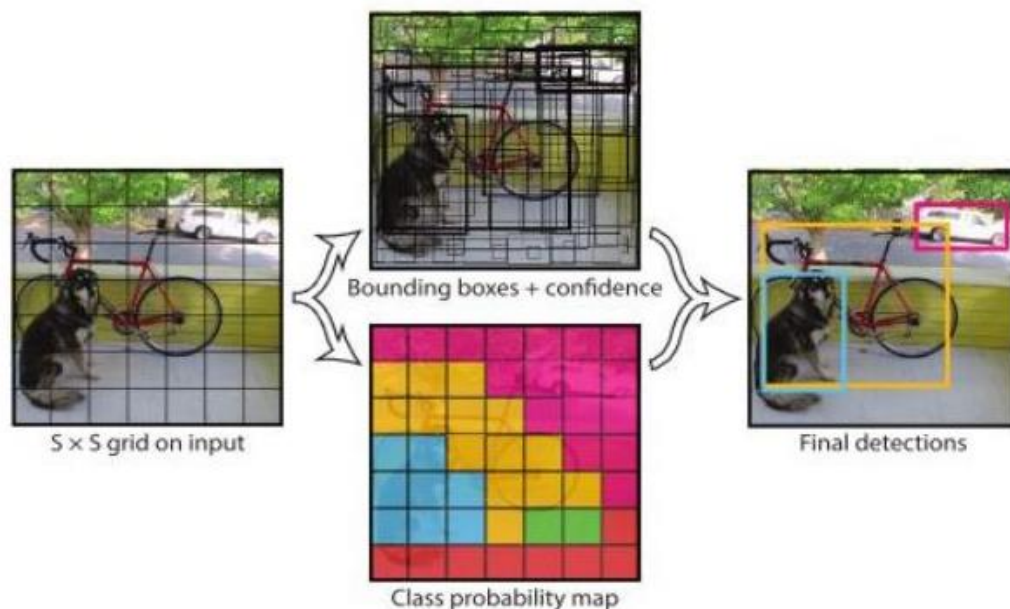
TensorFlow Lite memang merupakan antarmuka yang digunakan untuk mengekspresikan algoritma pembelajaran mesin dan untuk menjalankan perintah dengan menggunakan informasi yang dimiliki tentang objek atau target yang dikenali. Ini memungkinkan model pembelajaran mesin yang telah dihasilkan sebelumnya untuk dijalankan pada perangkat dengan sumber daya terbatas seperti perangkat seluler atau mikrokontroler [24]. Selain itu, TensorFlow, versi utama dari framework TensorFlow, memiliki kemampuan untuk melakukan pelatihan model pembelajaran mesin menggunakan CPU dan GPU. Penggunaan GPU dalam pelatihan model dapat menghasilkan waktu pelatihan yang lebih cepat dibandingkan dengan CPU, karena GPU memiliki arsitektur yang sangat cocok untuk tugas-tugas yang melibatkan operasi matriks paralel yang umum dalam pembelajaran mesin. Hal ini memungkinkan para peneliti dan praktisi untuk melatih

model yang lebih besar dan kompleks dalam waktu yang lebih singkat, meningkatkan produktivitas dalam pengembangan model pembelajaran mesin..

2.7. *You Only Look Once (YOLO)*

YOLO adalah sebuah algoritma yang dikembangkan khusus untuk mendeteksi objek secara real-time dalam citra atau video. Pendekatan yang digunakan oleh YOLO berbeda dengan pendekatan lain yang memerlukan beberapa tahapan, seperti mengidentifikasi objek pertama dan kemudian melakukan perbaikan. Sebaliknya, YOLO menggabungkan tahapan-tahapan tersebut menjadi satu proses yang dilakukan secara bersamaan [6].

Dalam YOLO, citra input dibagi menjadi grid berukuran $S \times S$. Dalam kasus yang Anda sebutkan, nilai S adalah 7, dan citra input berukuran 448×448 pixel. Setiap sel dalam grid bertanggung jawab untuk memprediksi beberapa bounding box (kotak pembatas) yang mungkin mengandung objek. Umumnya, ada 2 bounding box yang diprediksi dalam satu sel grid. Oleh karena itu, untuk setiap sel, kita memiliki $2 \times B$ bounding box, di mana B adalah jumlah bounding box yang diprediksi dalam satu sel biasanya 2 [25]. Proses kerja YOLO ditampilkan pada Gambar 2.1 berikut:



Gambar 2.1 Proses Kerja YOLO [25]

Berdasarkan Gambar 2.1 Dalam YOLO, normalisasi atribut pada bounding box diperlukan untuk memastikan bahwa semua nilai berada dalam rentang yang seragam, yaitu antara 0 hingga 1. Ini membantu dalam pelatihan dan pemrosesan selanjutnya. Normalisasi ini penting karena membuat representasi bounding box seragam dan independen terhadap resolusi gambar asli, sehingga memudahkan proses pelatihan dan inferensi. Dengan normalisasi ini, model YOLO dapat memprediksi bounding box dengan koordinat dan ukuran yang dapat diinterpretasikan dengan baik dalam berbagai ukuran citra [11].

2.8. Deep Learning

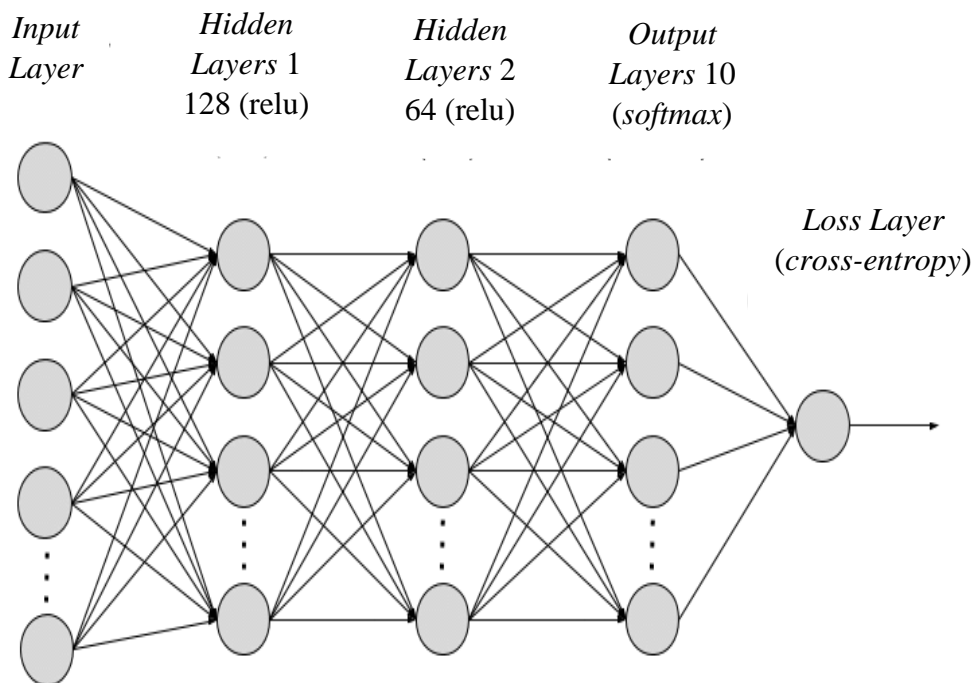
Teknologi deep learning memungkinkan komputer untuk memahami data yang lebih kompleks, yang sering kali sulit diproses dengan metode tradisional. Ini membuka peluang untuk berbagai aplikasi kecerdasan buatan yang dapat membantu dalam menjalankan tugas-tugas yang sebelumnya membutuhkan kecerdasan manusia. Beberapa contoh aplikasi deep learning yang umum digunakan termasuk:

1. **Pengenalan Gambar:** Deep learning digunakan untuk mengenali objek dalam gambar, seperti dalam sistem pengenalan wajah, deteksi objek, dan klasifikasi gambar.
2. **Penerjemahan Bahasa:** Model deep learning digunakan dalam sistem penerjemahan bahasa otomatis, yang dapat menerjemahkan teks dari satu bahasa ke bahasa lainnya.
3. **Pemrosesan Suara:** Deep learning digunakan dalam aplikasi pemrosesan suara, seperti pengenalan ucapan, transkripsi otomatis, dan asisten suara.
4. **Mendeteksi Pola:** Deep learning dapat digunakan untuk mendeteksi pola kompleks dalam data, seperti prediksi harga saham, analisis sentimen dalam teks, dan pemrosesan sinyal medis.
5. **Otomatisasi Tugas:** Model deep learning dapat digunakan untuk otomatisasi tugas-tugas yang repetitif, seperti penggantian teks dalam gambar atau mengkategorikan email spam.

Dengan kemampuan deep learning untuk mengenali pola yang rumit dalam data, teknologi ini telah menjadi sangat berharga dalam berbagai bidang, termasuk teknologi medis, otomotif, penerbangan, dan banyak lagi. Ini terus mendorong

perkembangan aplikasi kecerdasan buatan yang canggih dan memengaruhi berbagai aspek kehidupan [29].

Seperti otak manusia, deep learning juga menggunakan proses pembelajaran untuk mengatur bobot dan parameter di antara neuron-neuron dalam jaringan. Selama pelatihan, jaringan neural "menggunakan" data pelatihan untuk mengoptimalkan bobot-bobot ini sehingga dapat melakukan tugas tertentu, seperti pengenalan pola atau klasifikasi. Jaringan neuron buatan adalah algoritma *deep learning* yang menggunakan simpul ini untuk memecahkan masalah kompleks. Berikut adalah komponen jaringan *deep learning* pada Gambar 2.2.



Gambar 2.2 Komponen Jaringan *Deep Learning* [29]

Berdasarkan Gambar 2.2 dapat dilihat komponen jaringan *deep learning* memiliki beberapa lapisan diantaranya ialah Jaringan neural deep learning memiliki banyak lapisan tersembunyi, yang dapat bervariasi dalam jumlah tergantung pada arsitektur jaringan. Jumlah lapisan tersembunyi yang dalam atau kompleks dapat memungkinkan jaringan untuk menganalisis masalah dari berbagai sudut pandang, memungkinkan pembelajaran fitur yang lebih abstrak dan pemecahan masalah yang lebih rumit. Namun, penggunaan lapisan tersembunyi yang lebih dalam juga dapat meningkatkan kompleksitas dan kebutuhan komputasi dalam pelatihan jaringan.

Oleh karena itu, desain arsitektur jaringan deep learning harus mempertimbangkan trade-off antara kedalaman dan kinerja

Namun, dalam tugas klasifikasi yang lebih kompleks dengan banyak kategori yang berbeda misalnya, klasifikasi gambar hewan menjadi berbagai jenis hewan, jumlah simpul di lapisan output akan sesuai dengan jumlah kategori tersebut. Setiap simpul dalam lapisan output mewakili probabilitas atau skor kategori tertentu. Dalam kasus ini, model deep learning mencoba untuk menghasilkan prediksi yang lebih rinci dengan memberikan skor untuk setiap kategori yang mungkin, dan kategori dengan skor tertinggi akan menjadi prediksi akhir.

2.9. Scale Modifier

Scale modifier bertujuan untuk mengurangi atau mengecilkan ukuran dimensi dari gambar yang dipilih. Biasanya gambar yang di pangkas memiliki ukuran yang besar oleh karna itu diperlukan *scale modifier* untuk memunculkan ukuran gambar yang lebih kecil [26]. Efek menggunakan *scale modifier* adalah akan memperoleh tingkat akurasi yang lebih baik karna dapat mendeteksi gambar pada kondisi yang kecil sehingga YOLO dapat dengan lebih mudah mendeteksi objek yang sudah kita tentukan atau *training* terlebih dahulu datanya.

2.10. Optimasi Model

Optimasi model merupakan sebuah cara yang bertujuan mengurangi ukuran *file*, mengurangi latensi, dan mengkompabilitasi *akselerator*. Pada dasarnya, optimasi model bekerja dengan mengurangi ketepatan dari angka yang digunakan untuk merepresentasikan parameter model [27]. Efek yang terjadi saat mengoptimasi model ialah model ukurannya menjadi lebih sama dan cara untuk memperoleh hasil yang terbaik.

2.11. Google Colaboratory

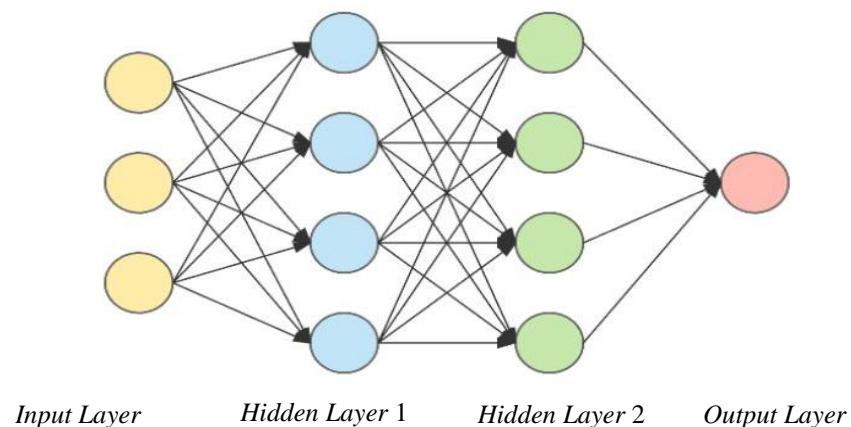
Google Colaboratory adalah sebuah proyek yang bertujuan untuk menyebarkan pendidikan dan penelitian *machine learning* [28]. Buku catatan kolaboratoris didasarkan pada Jupyter dan berfungsi sebagai objek Google

Dokumen dapat dibagikan dan pengguna dapat berkolaborasi di buku catatan yang sama. Colaboratory menyediakan *runtime* Python 2 dan 3 yang telah dikonfigurasi sebelumnya dengan pustaka *machine learning* dan *artificial intelligence* yang penting, seperti TensorFlow, Matplotlib, dan Keras. Mesin virtual di bawah *runtime* (VM) dinonaktifkan setelah jangka waktu tertentu, dan semua data dan konfigurasi pengguna hilang. Namun, *notebook* tersebut dipertahankan, dan juga memungkinkan untuk mentransfer *file* dari *hard disk* VM ke akun Google Drive pengguna [29].

Google Colaboratory mempunyai 4 virtual GPU yaitu Nvidia Tesla K80s, T4s, P4s, dan P100s. Hanya GPU Nvidia Tesla K80s yang dapat digunakan secara gratis, sisanya harus menggunakan Google Colaboratory yang versi berbayar. Dalam menggunakan Google Colaboratory yang gratis disediakan juga *memory* GPU sebesar 12 GB, RAM sebesar 25 GB, dan *temporary disk* sebesar 107 GB.

2.12. Artificial Neural Network

Artificial Neural Network bekerja dengan mengalirkan informasi dari *layer* masukan melalui *layer* tersembunyi ke *layer* keluaran. Hubungan antara neuron-neuron di seluruh jaringan memiliki bobot yang disesuaikan selama pelatihan jaringan untuk mengoptimalkan hasil prediksi [30]. Contoh penggambaran model *artificial neural network* dapat dilihat pada Gambar 2.3.




Gambar 2.3 Model Dasar *Artificial Neural Network* [31]

Berdasarkan Gambar 2.3 dapat dilihat model dasar jaringan syaraf tiruan memiliki lapisan *input* biasanya dalam bentuk vektor multidimensi ke lapisan *input*

yang akan mendistribusikannya ke lapisan tersembunyi, lapisan tersembunyi kemudian akan membuat keputusan dari lapisan sebelumnya dan mempertimbangkan bagaimana perubahan statistik di dalam dirinya dan meningkatkan hasil akhir, dan ini disebut sebagai lapisan *output*. Kemudian ada Jaringan Syaraf Tiruan (JST) yang neuron di dalam *layer*-nya tersusun menjadi 3 dimensi ialah *Convolutional Neural Network* (CNN) [31]. CNN telah mencapai kesuksesan besar dalam sejumlah tugas pengolahan citra, termasuk keberhasilan dalam kompetisi seperti ImageNet, di mana CNN telah menjadi arsitektur utama yang digunakan dalam klasifikasi gambar. Namun, kegunaan CNN tidak terbatas pada citra saja; mereka juga telah diterapkan dalam berbagai bidang pemrosesan data spasial, termasuk pemrosesan sinyal suara dan pemrosesan video.

2.13. *Intersection Over Union*

Intersection Over Union (IoU) adalah metrik yang penting dalam mengevaluasi kinerja model deteksi objek, dan sering digunakan dalam pelatihan model dengan mengukur sejauh mana prediksi bounding box mendekati ground-truth. Nilai IoU yang diharapkan dapat bervariasi tergantung pada tugas dan datasetnya, tetapi seringkali ada batasan IoU minimum yang didefinisikan sebagai ambang batas untuk menganggap deteksi berhasil atau tidak berhasil[32]. Ilustrasi persamaan IoU dapat dilihat pada Gambar 2.4.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Gambar 2.4 *Intersection Over Union* [32]

Berdasarkan Gambar 2.4 dapat dilihat persamaan *Intersection over Union* (*IoU*) adalah titik temu atas dan *union* hanyalah sebuah rasio. Di pembilang menghitung area *overlap* antara kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. Penyebutnya adalah area *union*, atau lebih

sederhananya, area yang dicakup oleh kotak pembatas yang diprediksi dan kotak pembatas kebenaran dasar. Menghitung melalui membagi area atas antara kotak pembatas dengan area *union* menghasilkan nilai akhir.

2.14. *Confusion Matrix*

Confusion matrix adalah tabel yang menggambarkan empat kondisi yang dapat muncul saat sebuah model klasifikasi memproses data[33]. *Confusion Matrix* dan metrik yang dihitung dari tabel ini sangat berguna dalam mengevaluasi sejauh mana model klasifikasi mampu melakukan tugasnya dan mengidentifikasi kekuatan serta kelemahan dari model tersebut Isi dari tabel *confusion matrix* ada 4, yaitu:

- a. True Positive (TP): Ini adalah kondisi di mana model mengklasifikasikan data sebagai positif (ya/True), dan kenyataannya juga positif (ya/True). Ini mengindikasikan bahwa model dengan benar mengidentifikasi contoh positif.
- b. True Negative (TN): Ini adalah kondisi di mana model mengklasifikasikan data sebagai negatif (tidak/False), dan kenyataannya juga negatif (tidak/False). Ini menunjukkan bahwa model dengan benar mengidentifikasi contoh negatif.
- c. False Positive (FP): Ini adalah kondisi di mana model mengklasifikasikan data sebagai positif (ya/True), tetapi kenyataannya negatif (tidak/False). Ini juga dikenal sebagai kesalahan Type I atau "false alarm."
- d. False Negative (FN): Ini adalah kondisi di mana model mengklasifikasikan data sebagai negatif (tidak/False), tetapi kenyataannya positif (ya/True). Ini juga dikenal sebagai kesalahan Type II atau "miss."

Berikut Persamaan yang akan digunakan berdasarkan data dari *confusion matrix* adalah akurasi, presisi, *recall* dan *f-score*.

1. Nilai akurasi mengukur sejauh mana model benar dalam klasifikasi secara keseluruhan. Nilai akurasi adalah metrik yang berguna untuk menilai seberapa baik model Anda dalam konteks umum. Namun, perlu diingat bahwa akurasi mungkin tidak selalu menjadi metrik yang paling informatif jika Anda memiliki ketidakseimbangan kelas dalam dataset. Dalam kasus tersebut, metrik lain seperti

presisi, recall, dan F1-Score dapat memberikan pandangan yang lebih baik tentang kinerja model Akurasi dapat dicari menggunakan Persamaan (2.1).

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \times 100\% \quad (2.1)$$

Berdasarkan Persamaan 2.1 nilai akurasi dapat dihitung dengan mencari nilai *true positive*, *true negative*, *false positive*, dan *false negative* antara kelas dan hasil prediksi dari modelnya.

2. *Precision* adalah mengukur sejauh mana model benar dalam mengklasifikasikan positif dan dihitung. Presisi mengukur tingkat keakuratan prediksi positif. Presisi dapat dihitung menggunakan Persamaan (2.2) berikut.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.2)$$

Berdasarkan Persamaan 2.2 nilai presisi dapat dihitung menggunakan hasil nilai dari *true positive* yang diberikan oleh model dibagi dengan hasil nilai dari *true positive* ditambah dengan hasil nilai *false positive*.

3. *F-Score* adalah rata-rata harmonik dari presisi dan recall dan digunakan untuk mengukur keseimbangan antara keduanya Mencari nilai *F-Score* dapat diketahui melalui persamaan (2.3) berikut.

$$\text{F-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (2.3)$$

Berdasarkan Persamaan 2.3 nilai *F-Score* didapatkan melalui perkalian nilai *precision* dan *recall* yang dikali 2 dan dibagi dengan nilai *precision* ditambah dengan hasil nilai dari *recall*.

4. *Recall* mengukur sejauh mana model dapat mendeteksi semua contoh positif. Recall dapat dicari nilainya menggunakan dengan Persamaan (2.4).

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

Berdasarkan Persamaan 2.4 nilai recall didapatkan melalui nilai true positive dibagi dengan nilai true positive lalu dijumlah dengan nilai hasil false negative.

2.15. *Computer Vision*

Computer vision adalah bagian dari ilmu komputer yang membahas bagaimana sebuah komputer dapat melihat seperti manusia, oleh karena sangat erat kaitannya dengan penglihatan, pencahayaan menjadi faktor yang juga penting dalam hal ini [34]. Pencahayaan adalah salah satu faktor penting dalam computer vision karena dapat memengaruhi kualitas citra yang dianalisis.

Dalam pengembangan computer vision, teknik dan algoritma yang kompleks digunakan untuk mendeteksi objek, melacak gerakan, mengenali wajah, melakukan segmentasi gambar, dan banyak tugas visual lainnya. Aplikasi dari computer vision sangat luas, mulai dari penginderaan mesin untuk otomatisasi pabrik hingga kendaraan otonom, pengawasan keamanan, pengenalan karakter optik, hingga pengolahan medis.

Dengan teknologi *computer vision*, kita dapat memantau pergerakan benda dalam waktu nyata dan mengestimasi kecepatannya. Ini bisa digunakan dalam berbagai konteks, termasuk lalu lintas jalan raya, pengukuran kecepatan dalam perlombaan olahraga, pengawasan pergerakan logistik di pabrik, dan banyak aplikasi lainnya.

Selain itu, computer vision juga telah membantu dalam pengembangan teknologi kendaraan otonom yang dapat melihat dan merespons dunia di sekitarnya, serta dalam berbagai bidang lainnya yang memanfaatkan data visual untuk pengambilan keputusan. Dengan kemajuan teknologi dalam computer vision, aplikasi yang lebih canggih terus dikembangkan untuk memecahkan berbagai masalah dunia nyata

2.16. Kajian Pustaka

Dalam penulisan ilmiah, penting untuk memberikan rujukan yang tepat pada penelitian sebelumnya. Ini dapat dilakukan melalui penulisan kutipan dan bibliografi yang merinci semua sumber yang digunakan. Dengan cara ini, penelitian baru dapat dihubungkan dengan kerangka pengetahuan yang sudah ada dan memberikan kontribusi yang berarti bagi bidang tersebut. Referensi berikut menjelaskan sistem deteksi objek wajah manusia. pengenalan individu melalui identifikasi wajah telah dilakukan dengan akuisisi video, anotasi label kelas, dan pembuatan model dengan metode YOLOv5. Model terbaik untuk pelatihan pengenalan wajah 70 mahasiswa diperoleh mAP 0,5 sebesar 99,4% dengan pengaturan parameter $epoch = 250$, $batch\ size = 16$, dan $learning\ rate = 0,01$. Hasil akurasi rata-rata terbaik adalah 99,88% yang ditunjukkan dengan pengujian pada video yang berisi satu siswa dengan tingkat kecerahan normal [35].

Penelitian yang lain menggunakan metode *You only look once* (YOLO) untuk pengenalan wajah, yang memiliki kecepatan mendeteksi dan akurasi yang tinggi. *You only look once* (YOLO) dalam deteksi wajah memiliki tingkat akurasi 100% dengan sudut kamera berada di posisi depan, kanan, dan kiri terhadap wajah, Untuk pengenalan dan identifikasi wajah dengan berbagai sudut pandang memiliki akurasi 100%. sedangkan pengujian pengenalan dengan jarak 30-100 cm memiliki akurasi 100% Dan jarak 5-20 cm dan memiliki akurasi 100% [36].

Selanjutnya ada penelitian tentang memanfaatkan kamera CCTV yang ada di setiap lantai untuk mendeteksi keberadaan orang yang ingin ditemui. menggunakan CNN (Convolution Neural Network) untuk mengolah data, metode yang digunakan adalah *object detection*, menggunakan algoritma YOLO (*You Only Look Once*) versi 4 dan OpenCV untuk mendeteksi manusia yang terlihat dan mengenalinya. Data yang digunakan berupa data foto, untuk setiap orang diambil 30 buah foto menghasilkan akurasi 80% [37].

Penelitian selanjutnya menggunakan *Convolutional Neural Network* (CNN) proses pendeteksian citra wajah digunakan Algoritma YOLO. Pendeteksian wajah dilakukan terhadap *input* berupa video hasil perekaman data dari *webcam* yang memuat ekspresi wajah, data ekspresi wajah dilatih dengan menggunakan CNN menghasilkan akurasi 85% dan akurasi *convolutional neural network* 46% [38].

Kemudian masih menggunakan metode *You Only Look Once* (YOLO) sebagai dalam pembuatan aplikasi deteksi jumlah orang pada area indoor ntuk mendukung pelaksanaan PPKM dengan metode YOLO dalam proses deteksi jumlah orang dengan jumlah data yang diuji 15 orang mendapatkan hasil rata-rata akurasi 86,7% [39].

Penelitian lainnya yang menggunakan metode *You Only Look Once* (YOLO) sebagai dalam perancangan alat identifikasi wajah dengan algoritma *You Only Look Once* (YOLO) untuk presensi mahasiswa. Prediksi wajah pada manajemen absensi akan dikirimkan ke sebuah *website monitoring* presensi mahasiswa. Hasil pengujian didapatkan rata-rata akurasi 0,9793 dengan memperhatikan parameter berupa pencahayaan dan *real time* mengirimkan ke *website* [40].

Penelitian selanjutnya yang menggunakan metode *You Only Look Once* (YOLO) adalah sistem pemantauan aktivitas keseharian lansia berbasis deteksi objek menggunakan algoritma YOLO. Data yang didapatkan akan dikirimkan ke *smartphone caregiver* dan keluarga via Telegram dalam bentuk pesan dan notifikasi. Dengan sistem tersebut, *user* dapat mengetahui aktivitas lansia dilokasi melalui aplikasi Telegram saat *user* di luar jangkauan lansia dalam bentuk pesan rutin berupa data aktivitas lansia. Hasil pengujian didapatkan rata-rata akurasi hasil Presisi 100%, *Recall* 100%, *F1 Score* 100%, *Average IoU* 87.26%, *Average Loss* 6.41%, *mAP* 100% serta akurasi yang dihasilkan mencapai 100% dengan parameter model yang digunakan adalah Rasio 90%:10%, *Batchsize* 64, *Learning rate* 0.008 dan *Max Batches* 4000 [8].

Penelitian yang lain menggunakan metode *You Only Look Once* (YOLO) adalah sistem absensi yang dapat dijalankan dengan fitur *Global Positioning System* (GPS) untuk secara otomatis mengecek lokasi pemilik wajah. Dari 20 data penilaian yang dilakukan sistem, sistem manajemen absensi dengan fitur pengenalan wajah dan GPS menggunakan YOLO pada *platform* Android menghasilkan akurasi sebesar 0.93435 dan terendah masih dalam range 93%, sedangkan nilai rata-rata akurasi adalah 93.26% [6].