

BAB II

DASAR TEORI

2.1 KAJIAN PUSTAKA

Berdasarkan Penelitian [1] dari Al Muhtadi Ambarak, dkk pada tahun 2023. Penelitian ini membuat aplikasi yang digunakan untuk mampu mendeteksi bahasa isyarat. Untuk konsep awal dari penelitian ini membuat studi literatur merancang *software* serta programnya, dimana program akan belajar dengan menggunakan konsep *machine learning* menggunakan *platform tensorflow lite*. Pada hasil data dibagi menjadi beberapa bagian yaitu *problem scoping*, data *acquisition*, data *preparation*, *modeling*, serta *evaluation*. Namun kekurangannya untuk pengembangan aplikasinya sendiri penelitian ini masih tahap pengembangan, dimana masih banyak proses yang harus diperbaiki dari si penulis. Penelitian ini bertujuan agar aplikasi yang dibuat mampu untuk membantu kaum tuna wicara dalam berkomunikasi dengan kaum dengar.

Berdasarkan riset [6] yang telah dilakukan oleh Pulung Adi, dkk pada tahun 2020. Pada riset ini membahas mengenai computer yang mampu untuk belajar sedemikian rupa layaknya manusia terutama dalam mengenali ekspresi dari wajah manusia, dengan menggunakan konsep dari machine learning. Program yang dibuat sedemikian rupa nantinya akan membaca serta mempelajari *dataset* melalui data *training* yang telah tersedia. dalam mendeteksi citra atau gambar metode yang paling signifikan serta akurat adalah menggunakan metode CNN. Tahapan dari metode penelitian yang digunakan pada jurnal ini adalah dengan menggunakan tahap *pre-processing*, serta tahap klasifikasi. Proses *training* yang terdapat pada program akan menggunakan *batch size* dan *epoch* yang memiliki nilai bervariasi. *Dataset* yang diambil sebanyak tujuh ekspresi manusia, yakni sedih, takut, jijik, senang, kaget, marah, dan netral. Dari hasil *training* yang didapat dengan menggunakan *epoch* 100, serta *batch size* 128 didapatkanlah hasil akurasi sebesar 90%. Dari total 35 ekspresi yang diberikan sebagai data *test*, program mampu menebak dengan benar dan akurat sebanyak 28 ekspresi dengan tingkat akurasi 80%. Perlu diketahui bawasanya jumlah dari *dataset* tidak sepenuhnya

mempengaruhi hasil akurasi saat dilakukan *testing*, namun kedetailan citra yang digunakan oleh *dataset* akan mempengaruhi hasil dari akurasi yang akan diberikan.

Berdasarkan penelitian [7] oleh Muhamad Rizky Fauzan, dkk pada tahun 2021. *Object detection* merupakan suatu teknik agar komputer bisa melihat dan mengidentifikasi objek yang ada baik dalam gambar ataupun video atau dapat diartikan juga sebagai perancangan suatu sistem agar sistem dapat melihat dan mengidentifikasi objek seperti manusia. Dalam pembuatan ini diperlukan data-data pendukung yang nantinya data-data tersebut akan dilatih agar sistem dapat mengenali suatu objek. Pada penelitian ini menggunakan konsep *image processing* yang merupakan teknik pengolahan sinyal oleh komputer dengan memasukkan beberapa citra. Didalam penelitiannya akan menggunakan bahasa pemrograman *python* yang berorientasi objek tingkat tinggi. Pendeteksian objek dilakukan untuk mendeteksi objek secara otomatis oleh sistem yang kemudian sistem dapat mengenali objek yang terdeteksi sesuai dengan *class* yang sudah dibuat sebelumnya. Salah satunya mengenai pendeteksian plat nomor kendaraan. Pendeteksian ini dilakukan untuk mempermudah pengecekan plat nomor dengan menggunakan sistem dengan menggunakan data yang sudah dilatih. Penelitian ini dilakukan untuk memodifikasi penelitian sebelumnya mengenai pendeteksian teks dan plat nomor kendaraan.

Berdasarkan penelitian [8] yang dilakukan oleh Tengku Cut, dkk pada tahun 2019. Pada penelitian ini bertujuan untuk mengimplementasikan computer vision agar mampu untuk mengenali pola bentuk wajah dengan menggunakan bantuan library OpenCV yang bertugas sebagai kamera penangkap gambar sesuai dengan dataset yang telah diberikan kedalam program. Peneliti menggunakan metode Computer Vision dikarenakan metode ini dikenal memiliki kecepatan serta keakuratan yang tinggi dalam membaca citra atau gambar. Ketika pengujian dilakukan diambil beberapa sampel pengujian yang berbeda seperti pengaruh citra ketika adanya pencahayaan, pengaruh jarak, kemiringan, wajah yang terhalang oleh objek yang berada didepannya, serta berdasarkan beberapa objek yang menyerupai wajah. Ketika melakukan uji coba terlihat bahwa hasil akurasi yang diberikan cukup akurat. Adapun peneliti selanjutnya diharapkan mampu untuk membuat program yang dapat meningkatkan lagi nilai akurasi dari pengenalan

wajah tersebut. Hasil nilai error yang tinggi perlu dilakukan perbaikan agar sistem dapat lebih akurat, sehingga sistem mampu untuk mengenali serta mendeteksi wajah dengan baik meskipun terdapat perbedaan yang cukup signifikan antara citra pada data testing dengan data training yang tersedia.

Berdasarkan penelitian [9] dari Indah Inayatul Arifah, dkk pada tahun 2022. Pada penelitian ini membuat model cerdas yang digunakan untuk dapat membaca serta mengenali bahasa isyarat yang digunakan para kaum tuna wicara dalam berkomunikasi menjadi sebuah tulisan atau *text*. Metode yang digunakan pada penelitian yang dilakukan adalah dengan menggunakan *YOLO (You Only Look Once)* serta *Convolution Neural Network (CNN)*. Untuk tahapan serta alur yang dipakai pada peneliti adalah pengumpulan data, olah gambar yang akan dilakukan oleh *CNN* serta *YOLO*. Pada tahap perancangan sistem *YOLO* nantinya akan bekerja untuk mengidentifikasi serta mengenali dari bahasa isyarat yang ditampilkan, kemudian metode *CNN* akan mengelompokkan jenis dari gerakan tangan yang diberikan. Adapun dataset yang dipakai berupa citra atau gambar serta video yang berpatok pada BISINDO. Ketika peneliti melakukan uji coba dari 5 objek gambar yang meliputi angka satu, dua, tiga, empat, dan lima dengan menggunakan dataset sebanyak 250 citra yang meliputi 200 data *training*, serta 50 data *testing*, terdapat beberapa gambar atau citra yang tidak terdeteksi sesuai dengan data train yang sudah tersedia. Hal ini disebabkan oleh pengaruh cahaya serta perubahan latar belakang, Sehingga menghasilkan nilai akurasi kurang dari 90%.

Berdasarkan penelitian [10] dari Calvin Gerald, dkk. Pada penelitian ini membahas dalam pendeteksian mobil. Sebenarnya sangat mudah bagi manusia untuk mengetahui serta mengenali jenis mobil berdasarkan gambar ataupun video. Namun bagi *computer* sangat sulit dilakukan apalagi adanya pengaruh cahaya, latar belakang, serta kompleksitas yang menjadi tantangan utama. Sehingga dipenelitian ini dibuatlah sistem yang mampu untuk membantu *computer* dalam mengenali jenis mobil tersebut. Di dalamnya berisi 16.185 citra atau gambar yang dimana terbagi atas 8.144 data *training* dan 8.041 data *testing*. Untuk alur dari pengenalan objek tersebut dimana program nantinya akan belajar mengenai dataset yang sudah tersedia, kemudian dengan menggunakan konsep *Computer Vision*, program akan menangkap sekaligus mengenali jenis dari mobil yang dideteksi secara *real time*

serta gambar. Metode yang digunakan didalamnya adalah *You Only Look Once* (YOLO) dan *Convolution Neural Network* (CNN) yang dapat digunakan dengan baik dengan sistem pendeteksian *repurpose classifier* atau *lozализer* untuk melakukan deteksi. *Convolution Neural Network* akan mendeteksi serta mengolah citra atau gambar yang tersedia. Pada uji deteksi dan pengenalan mobil pada penelitian ini didapatkan hasil akurasi sebesar 88,1% dengan nilai *epoch* yang diberikan 100. Yang menjadi kekurangan dalam penelitian ini masih belum dapat ditemukan solusi terbaik guna menaikan hasil akurasi diatas 90% yang diakibatkan oleh masalah atau *noise* yang diterima.

Berdasarkan riset [11] yang telah dilakukan oleh Muhammad Yusqi, dkk pada tahun 2023. Pada riset ini membahas terkait bagaimana peneliti mampu untuk membuat sistem pengenalan deteksi wajah agar dapat dimanfaatkan pada aplikasi, baik dengan menggunakan data citra atau gambar maupun dalam bentuk vidio. Adapun metode pendekatan yang dipakai adalah *one stage detector* dengan algoritma *You Only Look Once* (YOLO), serta untuk pross klasifikasi wajah dan non wajah akan menggunakan arsitektur *Convolution Neural Network* (CNN). Pada proses penelitian, peneliti melakukan 3 skenario percobaan diantaranya perbandingan dengan ukuran *grid* yang tersedia, perbandingan dengan ukuran dari citra atau gambar, serta perbandingan dari jumlah *epoch* yang diberikan. Dengan ukuran *grid* yang diberikan masing-masing 3x3 dan 7x7. Setelah dilakukan tiga tahap pengujian, dapat disimpulkan bahwa sistem pendeteksian dengan menggunakan YOLO serta dataset yang berjumlah 1920 citra *training* dan 480 citra *testing* mampu mendapatkan hasil terbaik, denga jumlah epoch yang digunakan sebanyak 150. Nilai dari akurasi yang dihasilkan pada saat pendeteksian dapat dipengaruhi oleh metode *Convolution Neural Network* (CNN) yang dikembangkan pada program serta keseimbangan data yang didapatkan.

Berdasarkan riset [12] yang telah dilakukan oleh Kiki Wahyuddin, dkk pada tahun 2023. Pada riset ini membahas terkait perkembangan teknologi yang sangat cepat dengan kemajuan serta kecanggihan teknologi yang sangat dapat dirasakan khususnya pada sistem keamanan. Sistem keamanan dibuat dengan tujuan agar menghindari adanya pembobolan ataupun pencurian data atau sistem. Dalam riset ini akan menggunakan opsi sistem keamanan dengan menggunakan biometric

wajah dengan menggunakan metode *Convolution Neural Network* (CNN). Pengenalan wajah yang dilakukan dengan menggunakan beberapa fitur seperti pengambilan gambar saat posisi yang berbeda, jarak pandang wajah dengan kamera, serta mimik atau ekspresi dari pengguna. Didalam pengujian terdapat 15 sampel wajah dengan percobaan sebanyak 30 kali. Dari hasil penelitian yang didapatkan dimana sistem kinerja dari pendeteksi wajah dengan kunci otomatis yang dibuat memperoleh tingkat akurasi yang belum maksimal yaitu sebesar 76% dalam kondisi pencahayaan gelap serta terang. Adapun saran untuk penelitian selanjutnya diharapkan mampu untuk membuat sistem deteksi wajah dengan alat yang memiliki spesifikasi yang tinggi, sehingga nilai akurasi yang didapatkan nantinya dapat maksimal tanpa harus dipengaruhi oleh jarak ataupun pencahayaan disekitar.

2.2 DASAR TEORI

Penelitian ini memiliki tujuan untuk mengembangkan sistem pendeteksi bahasa isyarat alfabet BISINDO dengan menggunakan sejumlah kerangka teori yang telah ditetapkan sebagai pedoman. Berikut beberapa teori yang dapat digunakan sebagai penunjang dalam proses perancangan pada kegiatan penelitian:

2.2.1 TUNA WICARA

Tuna wicara adalah kondisi dimana seseorang mengalami hambatan atau gangguan saat melakukan komunikasi secara verbal serta kesulitan untuk mendengar atau simpelnya tuna wicara merupakan ketidakmampuan seseorang dalam berkomunikasi atau berbicara dengan kaum dengar [13]. Mulai dari artikulasi hingga kefasihan dalam berkomunikasi merupakan alasan utama yang terletak pada gangguan dalam berbicara. Gangguan bicara juga dapat disebabkan oleh terhambatnya pertumbuhan janin saat di kandungan. Saat lahir, gejala dapat dilihat atau diidentifikasi oleh frekuensi pengulangan atau perpanjangan suara selama berbicara [14]. Seseorang yang mengalami tuna wicara biasanya menggunakan bahasa isyarat sebagai bahasa pengantar mereka dalam berkomunikasi dengan sesama. Adapun bahasa isyarat yang digunakan di Indonesia yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO).



Gambar 2. 1 Tuna wicara [14]

Pada Gambar 2.1 menunjukkan kondisi seseorang tuna wicara dalam berkomunikasi menggunakan bahasa isyarat dengan orang lain. Adapun gejala yang mendasari seseorang tersebut mengalami kondisi tuna wicara meliputi:

1. Ketika sedang berbicara, seseorang tersebut sering mengulangi kata-kata yang dia ingin sampaikan.
2. Suara terdengar serak, dan samar-samar.
3. Kesulitan dalam mengucapkan kata-kata dengan baik dan benar.

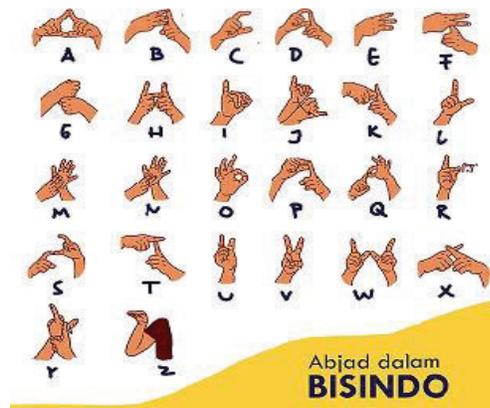
Secara fisik, seseorang dengan gangguan komunikasi atau tuna wicara ini dapat mengalami gejala-gejala sebagai berikut:

1. Terdapat cairan yang keluar dari alat pendengaran telinga.
2. Kondisi bibir yang menunjukkan kelainan bentuk seperti terdapat retakan.
3. Sering melakukan gerakan yang berulang-ulang.
4. Ketika sedang berbicara dengan sesama, mereka cenderung memperhatikan gerakan bibir lawan bicaranya.
5. Memiliki sifat yang pendiam dan sulit untuk berinteraksi dengan sesama.

2.2.2 BAHASA ISYARAT INDONESIA (BISINDO)

Bahasa Isyarat Indonesia (BISINDO) merupakan salah satu jenis bahasa isyarat yang digunakan oleh kaum tuna wicara dalam berkomunikasi yang dikembangkan oleh kumpulan komunitas tuna wicara Indonesia [15]. BISINDO bermula dari bahasa awal dan muncul secara alami dalam budaya Indonesia dan digunakan dalam kehidupan sehari-hari, dimana pada penggunaannya disesuaikan

dengan pemahaman serta latar belakang tanpa memberikan struktur imbuhan bahasa Indonesia.



Gambar 2. 2 Bahasa isyarat Indonesia [15]

Seperti pada Gambar 2.2 menunjukkan kumpulan gestur alfabet A sampai Z bahasa isyarat yang digunakan oleh masyarakat Indonesia. Bahasa isyarat ini dilakukan dengan memperagakan menggunakan tangan. BISINDO juga merupakan jenis komunikasi visual, yang dimana selain gerakan tangan, ekspresi juga menjadi peran besar dalam menghidupkan suasana percakapan. BISINDO sendiri termasuk pada sistem komunikasi yang dinilai cukup efektif dan praktis. Dikatakan praktis dan efektif karena bahasa isyarat ini dapat dipelajari bukan hanya untuk kaum tuna wicara namun juga pada kaum dengar atau semua orang. Dengan mempelajari bahasa isyarat dapat mengurangi hambatan dalam berkomunikasi antara kaum dengar dan kaum tuna wicara dalam berkomunikasi serta mampu mendukung lingkungan yang inklusif. Selain itu, bahasa isyarat juga dinilai mampu untuk membantu keseimbangan perkebangan dari otak kanan dan otak kiri serta meningkatkan kecerdasan otak.

2.2.3 ARTIFICIAL INTELLIGENCE (AI)

Kecerdasan buatan atau disebut sebagai *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang dibuat guna untuk memecahkan masalah yang memiliki keterkaitan dengan kecerdasan manusia. *Artificial Intelligence* juga menjadi faktor atau kunci penggerak revolusi industry 4.0 yang memberikan akses kemudahan di berbagai sektor [16]. *Artificial Intelligence* memiliki fungsi umum

yaitu mampu untuk menganalisis serta mengklasifikasikan data seperti layaknya otak manusia. AI juga mampu untuk memberikan solusi praktis serta efektif sehingga mempercepat dan mempermudah dalam penyelesaian pengerjaan masalah yang kompleks sekalipun dengan tepat di berbagai aspek bidang pekerjaan. Pada Gambar 2.3 menunjukkan AI yang terbentuk berdasarkan kumpulan banyaknya *neuron*. Tujuan diciptakannya kecerdasan buatan adalah untuk meniru sistem kerja otak manusia, seperti mesin yang mampu untuk belajar, mengambil keputusan, melakukan penalaran, serta memeriksa masalah. Terdapat 4 jenis sistem AI yaitu:

1. *Acting Rationally*, merupakan sistem kecerdasan buatan yang digunakan untuk dapat bertindak secara rasional.
2. *Thinking Rationally*, merupakan sistem kecerdasan buatan yang digunakan untuk mampu berpikir secara rasional.
3. *Thinking Humanly*, merupakan sistem kecerdasan buatan yang digunakan untuk mampu berpikir layaknya manusia.
4. *Acting Humanly*, merupakan sistem kecerdasan buatan yang digunakan untuk dapat bertindak layaknya manusia.



Gambar 2. 3 Artificial Intelligence [17]

Pada penelitian yang dilakukan oleh Bullock dilakukan sebuah perbandingan antara manusia dengan kecerdasan buatan dalam memecahkan suatu masalah. Hasilnya didapatkan bahwa kecerdasan buatan mendominasi dalam memecahkan suatu masalah dengan membutuhkan kemampuan analisa tingkat tinggi, serta menghasilkan tingkat ketidakpastian yang rendah [17]. Singkatnya, cara dari Artificial Intelligence (AI) ini adalah dengan menggunakan data yang

telah dimasukkan untuk dipelajari lebih lanjut menggunakan program yang telah tersedia pada sistem. Nantinya AI mampu untuk mengidentifikasi data, megolah data, menganalisis data, kemudian dapat menyimpulkan keputusan yang diambil dari masalah yang didapat dan dipelajari mesin. Semakin banyak data yang diberikan atau sebagai data belajar mesin, mampu untuk meningkatkan kemampuan dari *Artificial Intelligence* dalam mendapatkan hasil akurasi. *Artificial intelligence* bekerja sesuai dengan algoritma yang terdapat pada sistem komputer yang telah diberikan pada proses pembuatannya. Algoritma yang digunakan sebagai kerangka berpikir dari kecerdasan buatan dalam memproses berbagai jenis data yang ada. Algoritma yang digunakan pada *Artificial Intelligence* memerlukan banyak data dan kuat agar komputer dapat membedakan pola serta klasifikasinya yang baik. Dengan banyaknya data yang dipakai maka akan membentuk algoritma yang kompleks, yang akan menciptakan sistem mesin yang mampu untuk seolah-olah berpikir layaknya manusia, membuat keputusan, belajar, dan juga beradaptasi dengan sistem.

Adapun kelebihan yang dimiliki oleh *Artificial Intelligence* (AI) antara lain, sebagai berikut:

1. Dapat menyelesaikan suatu pekerjaan dengan cepat dan memiliki ketepatan akurasi yang tinggi.
2. Kecerdasan buatan memiliki sifat permanen serta tidak berubah.
3. Memiliki kapasitas penyimpanan data yang tanpa adanya batasan.
4. Memiliki kemampuan memecahkan masalah dengan tingkat kompleksitas yang tinggi.
5. *Artificial Intelligence* dapat digunakan tanpa adanya batasan waktu.

Selain memiliki kelebihan, *Artificial Intelligence* (AI) juga memiliki beberapa kekurangan antara lain, sebagai berikut:

1. Tidak memiliki kemampuan untuk mengembangkan pengetahuan seperti layaknya otak manusia.
2. Kecerdasan buatan hanya mampu mengembangkan pengetahuan bergantung pada sistem yang dibangun.
3. Tidak memiliki *Common Sense*, yang dimana merupakan kemampuan yang bukan hanya sekedar memproses informasi, namun juga mengerti akan

informasi yang diberikan. Sedangkan AI tidak mampu untuk mengerti informasi.

4. Kecerdasan buatan yang terdapat pada AI bergantung pada program yang dibangun oleh seorang programmer.

2.2.4 *OPENCV*

OpenCV (Open Source Computer Vision Library) merupakan *library open source* yang dikembangkan oleh instansi dari *Intel Research* pada tahun 1999 sebagai kelanjutan dari proyek aplikasi berbasis CPU, *real-time tracing*, serta penampilan 3 dimensi. Gambar 2.4 merupakan logo dari library *OpenCv*. Kata *Open* dari kata *OpenCV* menandakan bahwa ini merupakan *open source* yang bersifat gratis atau tidak berbayar dalam penggunaannya. Sedangkan CV atau *Computer Vision* yang berarti dapat menggunakan komputer dalam mengolah citra atau gambar.



Gambar 2. 4 *OpenCV* [18]

Jadi, *OpenCV* digunakan untuk melakukan berbagai operasi kegiatan yang berkaitan dengan sistem *Computer Vision* seperti mengolah gambar atau citra secara *real time* [18]. Di dalam *OpenCV* sudah terdapat 2500 algoritma yang telah dioptimalkan, di dalamnya sudah termasuk algoritma untuk *Machine Learning* (ML) serta *Computer Vision* (CV) [7]. *OpenCV* dapat digunakan diberbagai bahasa pemrograman seperti *Java*, *Python*, *C*, *C++*, dan sebagainya [6]. Contoh penerapan *OpenCV* di lingkungan sekitar adalah pemasangan kamera yang berada pada parkir untuk membaca plat nomor kendaraan. Dari hasil gambar plat nomor yang

diambil nantinya gambar atau citra tersebut akan diolah dalam bentuk digital lalu menjadi data yang dapat dijadikan sebuah informasi.

OpenCV memiliki beberapa fitur utama yang dapat dimanfaatkan antara lain, sebagai berikut:

1. *Image* dan Video I/O, dengan menggunakan fitur ini *OpenCV* mampu untuk membaca data berupa gambar ataupun video dari *file*, serta juga dapat menciptakan *file* gambar atau citra ataupun video.
2. *Computer Vision* (CV) sering digunakan dalam pencitraan digital untuk API rendah dan menengah. API atau *Application Programming Interface* adalah penerjemah komunikasi antara klien dengan *server* atau pengguna untuk dapat menyederhanakan implementasi dari *software*. Atau dapat dikatakan merupakan sekumpulan perintah atau program yang digunakan untuk dapat membangun perangkat lunak untuk sistem operasi. Pada *Computer Vision*, *OpenCV* dapat mendeteksi garis, tepi, puncak, proyeksi elip, piramida gambar, pemasangan model, dan masih banyak lagi.
3. *Sampling* gambar dan transformasi, *OpenCV* digunakan untuk substraksi *subregion* dari gambar atau citra yang telah disediakan, *random sampling*, dan juga melakukan rotasi gambar. Metode ini digunakan ketika gambar atau citra berbentuk 3 dimensi.
4. Modul *computer vision high level*, dalam penerapannya pada *OpenCV*, *library* ini mampu untuk mendeteksi wajah, pengenalan wajah, termasuk dengan *optical flow*, sehingga digolongkan dalam tahap *high level*.
5. Metode untuk *Artificial Intelligence* dan *Machine Learning*, ketika menggunakan aplikasi berbasis *machine learning* maka dibutuhkan metode dari kecerdasan buatan, didalam *OpenCV* semua metode tersebut sudah lengkap dan dapat langsung digunakan.

Adapun kelebihan yang dimiliki oleh *library OpenCV* yaitu sebagai berikut:

1. Pada *OpenCV* memiliki *library* dokumen dengan jumlah yang banyak.
2. *OpenCV* mampu bekerja dengan cepat pada komputer yang sudah menggunakan *processor intel*.
3. Jika dibandingkan dengan menggunakan *software matlab* yang memiliki fungsi yang sama dengan *OpenCV* atau sama-sama mampu untuk mengolah gambar

atau citra, maka komputasi yang terdapat pada *OpenCV* jauh lebih ringan dibandingkan *Matlab*.

4. *OpenCV* difasilitasi dengan kemampuan pengolahan citra secara *real time*.

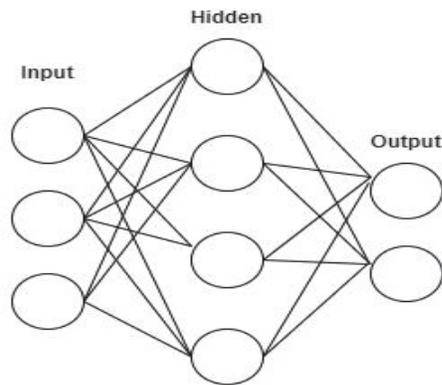
Adapun kekurangan yang dimiliki oleh *library OpenCV* yaitu sebagai berikut:

1. Pada *OpenCV* kurang mendukung dengan bahasa pemrograman yang berbasis .NET seperti *C#* dan *VB (Virtual Basics)*. Sehingga jika tetap ingin menggunakan bahasa pemrograman yang berbasis .NET mengharuskan untuk menggunakan *wrappers* atau entitas yang dapat membungkus item lain.
2. Instalasi yang lumayan ribet.

2.2.5 NEURAL NETWORK

Jaringan neural (Jaringan Syaraf Tiruan) merupakan salah satu metode yang terdapat pada kecerdasan buatan atau *Artificial Intelligence (AI)* yang diletakkan pada komputer sehingga komputer tersebut dapat belajar untuk memproses data secara benar dan akurat sama halnya dengan sistem cara kerja otak manusia [19]. Jaringan Neural ini merupakan tipe dari *Machine Learning (ML)*, yaitu *Deep Learning (DL)*. Pada konsepnya jaringan neural nantinya memiliki banyak simpul atau neuron yang saling terikat dan terhubung satu sama lainnya membentuk beberapa lapisan yang menyerupai lapisan otak manusia. Jaringan neural mampu untuk bekerja secara terus-menerus sehingga komputer dapat belajar dari kesalahan dan memperbaikinya. Proses belajar yang dimiliki jaringan syaraf tiruan ini digolongkan menjadi dua bagian yaitu belajar dengan pengawasan (*Supervised Learning*) dan belajar tanpa pengawasan (*Unsupervised Learning*) [20].

Oleh karena itu, jaringan neural dibuat agar mampu untuk memecahkan masalah yang rumit sekalipun tanpa henti karena sifat sistemnya yang selalu belajar, sehingga terciptalah sistem yang mampu untuk menyelesaikan masalah serta mendapatkan nilai akurasi yang tinggi. Jaringan neural dapat dikategorikan sebagai jaringan yang penting pada sistem dikarenakan membantu komputer dalam membuat keputusan yang cerdas tanpa bantuan manusia. Keputusan yang cerdas yang akan didapatkan tergantung pada data *input* serta *output* yang dimasukkan pada sistem.



Gambar 2. 5 Struktur lapisan neural network [20]

Pada Gambar 2.5 merupakan struktur lapisan *neural network*. Didalam jaringan neural terdapat 3 lapisan jaringan yang saling terhubung yaitu:

1. Lapisan *input*, pada lapisan ini merupakan lapisan terluar pada jaringan neural dimana setiap data yang akan masuk pertama kali akan melewati lapisan *input*. Lapisan *input* ini bekerja sebagai lapisan yang dapat memproses, menganalisis, serta mengelompokkan data yang masuk sehingga data tersebut nantinya dapat diteruskan ke lapisan jaringan neural berikutnya.
2. Lapisan tersembunyi, pada lapisan tersembunyi nantinya data yang terdapat pada lapisan *input* akan masuk ke lapisan ini. Lapisan tersembunyi memiliki jumlah yang besar. Lapisan tersembunyi ini memiliki peran penting dimana seluruh data dari lapisan *input* yang dikerjakan akan diproses sebagaimana sistem dibuat, sehingga nantinya jika data yang sudah diproses pada lapisan ini dapat diteruskan ke lapisan akhir jaringan neural.
3. Lapisan *output*, merupakan jaringan akhir atau lapisan akhir pada jaringan neural. Lapisan ini nantinya akan memberikan *output* sesuai dengan apa yang diminta atau memberikan hasil akhir pada sistem. Di lapisan akhir memiliki 1 atau beberapa simpul. Memiliki 1 simpul ketika output yang diberikan terdapat 2 kondisi akhir yaitu ya atau tidak, namun jika menggunakan klasifikasi multi kelas maka lapisan ini memiliki lebih dari 1 simpul.

Adapun kegunaan dari jaringan neural pada berbagai kasus penggunaan di sektor industri antara lain, sebagai berikut:

1. Jaringan neural dapat digunakan pada bidang kesehatan yang dimana mampu untuk mendiagnosis medis dengan menggunakan klasifikasi atau pengelompokan citra.
2. Jaringan neural mampu untuk mengolah keuangan dengan melakukan proses data historis.
3. Jaringan neural dapat memprediksi beban dari listrik yang dibutuhkan melalui kebutuhan permintaan energi.
4. Jaringan neural mampu untuk mengidentifikasi atau mengelompokkan senyawa kimia.

Selain kegunaan yang dimiliki jaringan neural, jaringan ini memiliki beberapa aplikasi penting yaitu sebagai berikut:

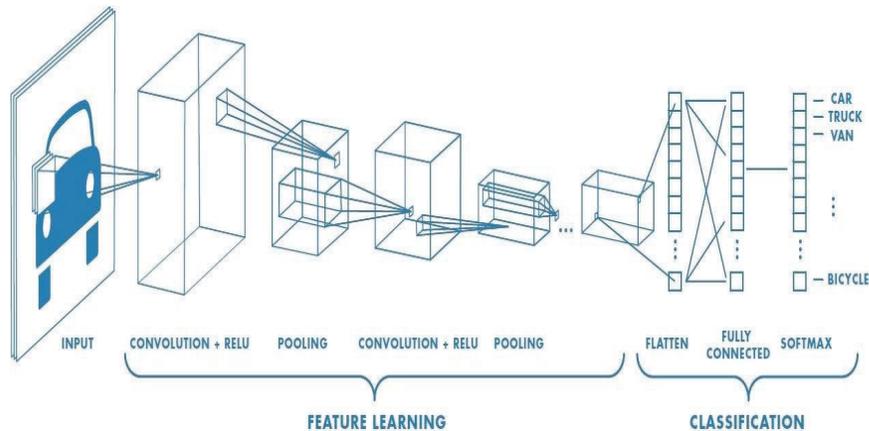
1. Penglihatan komputer, berdasarkan kata melihat berarti menunjukkan bahwa dengan menggunakan jaringan neural, komputer mampu untuk membedakan, mengelompokkan, serta mengenali gambar atau citra serta video yang disediakan. Contoh dari pengaplikasian jaringan neural untuk penglihatan komputer, seperti berikut ini :
 - Pengenalan wajah dengan mengidentifikasi wajah serta menganalisis atribut seperti mata tertutup dan mata terbuka.
 - Jaringan neural mampu untuk melabeli citra atau gambar berdasarkan logo merek, pakaian, dan detail dari citra lainnya.
 - Pengenalan visual yang diletakkan pada mobil kemudian otomatis dalam mengenali rambu-rambu lalu lintas.
2. Pengenalan suara, jaringan neural memiliki kemampuan untuk mengenali suara serta menganalisis ucapan manusia dengan menggunakan beberapa pembeda seperti tinggi rendahnya nada saat berbicara, cara pengucapan serta pelafalan dalam berbicara, dan lain-lainnya.
3. Pemrosesan bahasa alami, yang dimana merupakan proses teknologi dari machine learning yang memberikan kemampuan pada komputer dalam memanipulasi, memahami bahasa manusia. Nantinya dengan bantuan jaringan neural dapat membantu komputer untuk mengumpulkan wawasan serta makna teks pada dokumen yang disediakan.

2.2.6 CONVOLUTION NEURAL NETWORK (CNN)

Convolution Neural Network biasa disingkat CNN ataupun *ConvNet* merupakan golongan pada *Artificial Neural Network* (ANN) yang mampu untuk mengolah data citra atau gambar dalam melakukan pembelajaran untuk representasi fitur internal dan generalisasi fitur dalam masalah grafis umum seperti identifikasi objek dan masalah tampilan komputer lainnya [10]. *Convolutional Neural Network* (CNN) merupakan salah satu jenis *neural network* yang biasa digunakan pada data *image*. Pengolahan data yang terdapat pada *Convolution Neural Network* mengacu pada proses pendeteksian serta pengenalan objek pada sebuah gambar. *Convolution Neural Network* (CNN) dirancang untuk mengolah data dua dimensi [6]. CNN juga termasuk dalam metode *Deep Neural Network* dikarenakan memiliki kedalaman lapisan jaringan yang sangat tinggi, yang dimana jaringan ini terinspirasi oleh jaringan syaraf manusia [21]. *Convolution Neural Network* dapat mencakup ratusan lapisan, dan setiap lapisnya akan dilatih untuk mengenali aspek yang berbeda-beda dari sebuah gambar. Sama halnya dengan *deep learning*, *convolution neural network* ini pada tahap implementasinya dapat digunakan dengan data yang berukuran cukup besar, namun mengharuskan untuk memiliki perangkat keras yang memadai atau menyanggupi dalam beban komputasi saat sistem melakukan *training data*.

Convolution Neural Network ini juga tersusun sedemikian rupa membentuk sebuah filter dengan komponen panjang serta tinggi. Secara garis besar, proses kerja CNN ini yaitu dengan memanfaatkan konvolusi dengan meggerakan sebuah filter yang menyerupai bentuk kotak berukuran tertentu yang diarahkan pada objek atau gambar yang sudah disediakan. Dari hasil tersebut, nantinya akan memperoleh informasi baru. Filter tersebut dapat dipengaruhi oleh komplektifitas dari kualitas gambar seperti kecerahan, tepian, sampai dengan fitur yang lebih baik yang dapat mencirikan item tersebut. Ketika terdapat suatu gambar atau citra yang masuk pada sistem, nantinya akan memecah gambar menjadi gambar dengan ukuran lebih kecil dengan konvolusi yang sama. Selanjutnya gambar dengan ukuran yang kecil tadi akan dimasukkan ke *small neural network* untuk menghasilkan sebuah representasi fitur. Dengan tahap ini akan memberikan kemampuan pada *Convolution Neural Network* dalam mengenali sebuah objek. Hasil dari tahap sebelumnya akan

menyimpan gambar ke dalam *array* baru, kemudian terdapat proses *downsampling* atau *maxpolling* yang digunakan apabila suatu gambar atau citra tersebut memiliki nilai *array* yang masih terlalu besar dan harus dikecilkan. Dan tahap terakhirnya melakukan prediksi. Prediksi dilakukan pada *layer* terakhir yaitu *layer output* guna memberikan keputusan atau jawaban akhir. Adapun jenis arsitektur yang ada di *Convolutional Neural Network* (CNN) yaitu antara lain, *AlexNet*, *VGG16*, *VGG19*, *GoogleLeNet*, *Inception-V3*, *ResNet50*, *ResNet101*, *InceptionResNetV2*, *Squeezenet*, dan *MobileNet*.



Gambar 2. 6 Lapisan arsitektur CNN [21]

Gambar 2.6 menunjukkan lapisan arsitektur yang digunakan pada algoritma CNN. Arsitektur dalam *Convolution Neural Network* (CNN) ini dibagi menjadi 2 yaitu sebagai berikut:

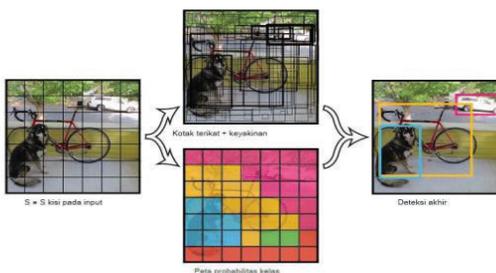
1. *Feature Extraction Layer*, proses pada bagian ini adalah *encoding*, dimana citra menjadi angka-angka yang mewakili citra tersebut. Lapisan ekstraksi fitur terdiri dari dua bagian, yaitu lapisan konvolusi dan lapisan *pooling*. Namun terkadang ada kajian atau esai yang tidak menggunakan *pooling*.
 - *Convolution Layer* digunakan untuk menghitung output atau keluaran dari jaringan neuron yang terhubung kedalam lapisan input.
 - *Rectified Linear Unit* (ReLU) mampu untuk menghilangkan vanishing gradient dengan menggunakan fungsi aktivasi element. Adapun kelebihan menggunakan ReLU yaitu mampumempercepat gradient gambar karena ReLU berbentuk linear. Dapat melakukan pembuata matriks ketika ambang batas berada pada nilai 0. Namun kekurangan dari penggunaan ReLU yaitu

sistemnya yang rapuh ketika data pada tahap training serta apabila citra atau gambar yang diberikan memiliki gradient yang cukup besar yang dapat menyebabkan *update* bobot pada ReLU.

- *Polling layer* merupakan *layer* yang dikenal dengan kecepatannya dalam komputasi dikarenakan parameter yang terus harus diperbaharui semakin sedikit, dan mampu untuk mengatasi masalah overfitting.
2. *Classification*, proses pada bagian ini adalah klasifikasi yang dilakukan sistem pada citra atau gambar yang tersedia.
- *Flatten* digunakan untuk membuat bentuk ulang dari sebuah citra menjadi *vector* agar dapat digunakan sebagai *input* ketika memasuki *layer fully connected*.
 - *Fully-Connected* (FC) digunakan untuk menghitung skor tiap kelas pada *dataset*.
 - *Softmax* dapat digunakan untuk menghitung probabilitas atau peluang ataupun kemungkinan pada suatu masalah dari setiap kelas yang diberikan.

2.2.7 OBJECT DETECTION

Object detection merupakan perancangan dari sistem yang dibangun yang bertujuan agar sistem mampu untuk mengenali serta mengidentifikasi objek baik dalam bentuk gambar maupun video. Seperti Gambar 2.7 menunjukkan *Object detection* didalam pembuatannya nanti akan dibangun berdasarkan data pendukung, sehingga sistem akan belajar serta dilatih dari data *training* yang telah disediakan untuk dapat mengenali suatu objek, dan dapat juga digunakan untuk memberikan informasi yang lebih baik dan banyak dari gambar yang tersedia serta membantu dalam pelabelan kelompok saat dilakukan pengkategorian citra [22].



Gambar 2. 7 Implementasi *object detection* [22]

Object detection juga mampu untuk mengembalikan informasi kelas tiap objek. Sistem kerja dari *object detection* ini akan digolongkan menjadi dua bagian atau kategori konsep, yaitu mana yang objek dan mana yang bukan objek. Ketika menggunakan sistem deteksi ini proses kerjanya terdapat dua jenis yaitu *soft detection* yang digunakan untuk mengenali suatu objek, dan *hard detection* yang digunakan untuk dapat mengenali objek serta posisi dari objek tersebut.

Pemanfaatan dari *object detection* yang paling sering dijumpai di lingkungan sekitar adalah pendeteksian plat nomor kendaraan pada area parkir secara *real time*. Dengan bantuan kamera, nantinya dapat mengenali objek berupa plat motor yang dapat dideteksi dan gambar akan diolah ke dalam bentuk digital [7]. Semisal ketika sistem diberikan *task* untuk mendeteksi objek, maka gambar objek tersebut digunakan sebagai data *input* berupa citra atau gambar. Selanjutnya gambar tersebut akan diteruskan kedalam sebuah model dari sistem yang sudah dilatih sebelumnya dengan menggunakan data *training* yang ada. Ketika melakukan data *testing*, sistem akan memberikan hasil atau *output* berupa posisi atau koordinat dari objek gambar yang tadi dimasukkan ke dalam sistem. Di dalam sistem *object detection* terdapat beberapa model yang dapat digunakan untuk mendeteksi suatu objek seperti *YOLO*, *Faster R-CNN*, *MobileNet*, *Mask R-CNN*, dan lain-lainnya.

2.2.8 DEEP LEARNING (DL)

Deep Learning (DL) merupakan teknik yang tergolong pada metode *Machine Learning* (ML) yang mampu untuk mengarahkan sebuah sistem komputer untuk dapat bekerja layaknya otak manusia. *Deep learning* adalah jenis pembelajaran data yang menggunakan beberapa lapisan pemrosesan untuk membangun representasi data. *Deep Learning* (DL) juga merupakan kunci perkembangan teknologi yang menggunakan kecerdasan buatan atau *Artificial Intelligence* (AI) dengan menggunakan algoritma yang mampu menghasilkan data yang akurat. Pada konsep dasarnya sebuah komputer atau mesin yang menggunakan *Deep Learning* nantinya akan mempelajari berbagai model masalah serta mengklasifikasikan tugas-tugas dari data yang sudah dikumpulkan ke sistem. Data yang dapat dikerjakan oleh *deep learning* dapat berupa gambar, suara, hingga teks [23]. Dengan menggunakan konsep *Deep Learning* biasanya mampu untuk

mengolah data yang berjumlah besar dan dapat menghasilkan tingkat akurasi yang tinggi. *Deep Learning* sangat marak digunakan pada kalangan *programmer* karena dirasa cukup untuk memudahkan aktivitas sehari-hari dengan bantuan mesin tanpa campur tangan manusia dengan menciptakan pekerjaan yang jauh lebih optimal dan efektif.

Cara kerja dari *Deep Learning* yaitu menggunakan metode *Neural Network*, sehingga terkadang sering didengar dengan sebutan *Deep Neural Network* yaitu bekerja dengan menggunakan lapisan atau *layer* yang berjumlah cukup besar yang merupakan salah satu keunggulan dari metode *Deep Learning*. Dimana metode ini mampu untuk mempelajari masalah dengan tingkat kompleksitas yang tinggi dan mampu memberikan hasil yang lebih akurat dibandingkan metode *Machine Learning Traditional*. Selain itu memiliki kemampuan yang dapat memberikan fitur-fitur yang dinilai sangat penting secara otomatis pada data yang diberikan sehingga pekerjaan akan lebih efisien dan tidak memakan banyak waktu. Namun dibalik dari kelebihanannya, juga terdapat kekurangan dimana dengan dapatnya memproses data dengan jumlah yang sangat besar, harus memiliki *hardware* atau alat pendukung yang memadai, serta dengan adanya interpretasi model yang sulit. Adapun jenis-jenis algoritma pada *Deep Learning* (DL) antara lain:

1. *Deep Neural Network* (DNN), merupakan struktur algoritma yang terdapat pada *deep learning* yang memiliki banyak lapisan atau *layer* dengan model sirkuit saraf manusia yang dibuat serta dirancang agar sistem mampu untuk mengenali pola yang disebut juga dengan *neural network* atau jaringan saraf tiruan.
2. *Artificial Neural Networks* (ANN), merupakan jaringan saraf tiruan dengan model yang kompleks yang digunakan agar sistem mampu untuk memprediksi keadaan ekosistem terhadap perubahan variabel lingkungan sekitar. Cara kerja dari *Artificial Neural Networks* ini memiliki pola kerja yang sama dengan cara kerja otak manusia.
3. *Convolutional Neural Network* (CNN), merupakan jenis algoritma yang terdapat pada *deep learning* yang didalam pengimplementasiannya paling sering digunakan dibandingkan algoritma yang lainnya. Didalam algoritma *Convolutional Neural Network* mampu untuk mengenali serta mengklasifikasi

gambar atau citra dengan memberikan hasil yang lebih canggih dalam memecahkan suatu masalah.

Pada dasarnya perbedaan yang terletak antara *Deep Learning* (DL) dan *Machine Learning* (ML) terdapat pada sistem pelabelan data seperti gambar, video, text. Yang dimana ketika menggunakan *Deep Learning* (DL) sistem pelabelan akan diberikan secara otomatis, namun ketika menggunakan *Machine Learning* (ML) sistem pelabelannya dilakukan secara manual. Berikut merupakan beberapa perbedaan yang terletak anatar *Machine Learning* (ML) dengan *Deep Learning* (DL) yaitu:

1. Penggunaan algoritma yang dilakukan oleh *Machine Learning* cenderung memiliki sifat mengurangi data di beberapa bagian, kemudian dari bagian-bagian yang telah dipisah tadi akan digabung menjadi satu untuk menghasilkan solusi atau *output*. Sedangkan pada konsep *deep learning*, konsep ini akan melihat secara keseluruhan masalah dalam satu proses. Contohnya ketika menggunakan *machine learning* maka akan melakukan proses deteksi objek dan pengenalan objek, sedangkan pada *deep learning* hanya dengan melatih program mampu untuk mengembalikan objek yang diidentifikasi dalam gambar ke dalam satu hasil.
2. Sistem *deep learning* akan memerlukan banyak waktu dalam proses pengerjaannya dibandingkan dengan *machine learning*. Ini disebabkan oleh banyaknya parameter serta rumus matematika yang rumit.
3. Dengan menggunakan banyak data yang dilatih atau dalam jumlah yang cukup besar, maka dengan menggunakan konsep *deep learning* seorang *programmer* harus mempunyai perangkat yang jauh lebih kuat dibandingkan dengan menggunakan konsep *machine learning*.

Di dalam konsep *Deep Learning* digunakan beberapa metode parameter ataupun pendekatan antara lain sebagai berikut:

1. *Hidden Layer*, parameter ini digunakan untuk menyatakan jumlah *layer* yang tersembunyi yang terdapat pada tiap *layer* untuk memberikan desain yang optimal dalam arsitektur Deep Learning (DL).
2. *Epoch*, digunakan untuk menunjukkan jumlah banyaknya iterasi yang harus digunakan pada saat proses pengumpulan data. *Epoch* juga merupakan tahapan

atau alur yang digunakan untuk mempelajari kumpulan data dengan pelatihan secara menyeluruh. Ketika algoritma *Deep Learning* telah menyelesaikan satu *epoch* maka sistem telah memperoleh pengetahuan dari *dataset training* secara menyeluruh.

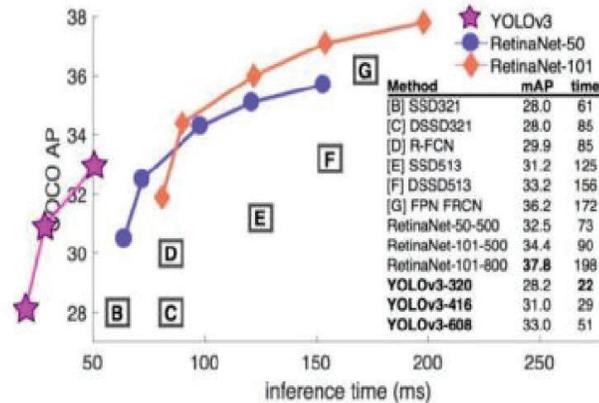
3. *Learning rate*, merupakan parameter yang digunakan untuk dapat menentukan nilai koreksi bobot. Nilai dari angka untuk tingkat pembelajaran adalah anatar nol hingga satu. Ketika *learning rate* dinaikkan, maka nilai akurasi yang didapatkan akan turun, namun jika *learning rate* diturunkan, maka akan menghasilkan nilai akurasi yang lebih tinggi, namun akan menjadikan proses *training* pada *dataset* atau pelatihan memakan banyak waktu.
4. *Evaluate Performance*, merupakan parameter performa sistem yang diukur berdasarkan nilai *accuracy*, sensitivitas, serta kurva *Receiver Operating Characteristic* (ROC). Performa dari model berdasarkan sistem yang telah dibuat dapat dipengaruhi oleh *batch size*, *epoch*, dan *learning rate*.

2.2.9 **YOU ONLY LOOK ONCE (YOLO)**

YOLO (*You Only Look Once*) merupakan salah satu model dari metode *deep learning* yang diusulkan pertama kali oleh Josep Redmond pada tahun 2015, dapat digunakan untuk mendeteksi objek [24]. Tugas dari YOLO sebagai pendeteksi objek adalah untuk mengetahui serta mampu untuk menentukan lokasi pada gambar dimana nantinya gambar tersebut akan diklasifikasikan oleh sistem. Hingga saat ini, *You Only Look Once* (YOLO) sudah memiliki beberapa versi yang terus ditingkatkan performanya mulai dari versi pertama hingga versi terbarunya. Pada penggunaannya, YOLO dinilai sangat efektif untuk mendeteksi objek dikarenakan metode ini cukup kecil dan ringan jika dijalankan pada perangkat komputasi atau komputer sekalipun. Pada Gambar 2.8 menunjukkan arsitektur yang digunakan oleh algoritma YOLO adalah dengan menggunakan *Convolution Neural Network* yang dimana memiliki 24 *Convolution layer* atau lapisan dan terdapat 2 *fully connected layers*. Fungsi dari *convolution layers* ini digunakan YOLO untuk dapat memperoleh fitur dari gambar atau citra yang dimasukkan, sedangkan kegunaan dari *fully connencted layer* adalah untuk memprediksi probabilitas dari hasil keluaran *output* dan koordinat citra.

2. Sistem kinerja yang dimiliki YOLO tergolong cepat dikarenakan YOLO memiliki arsitektur *framework* yang menggunakan model regresi kompleks.
3. YOLO dapat memberikan fitur untuk mempertimbangkan elemen global yang didapatkan dari citra yang sudah diprediksi. Kemudian mampu untuk menyandingkan informasi kontekstual dari citra tersebut.
4. YOLO sangat mudah diterapkan dan digunakan jika ingin memasukkan gambar baru ke dalam sistem.

Sementara itu, YOLO juga memiliki kekurangan yaitu walaupun YOLO dikenal dengan kecepatannya dalam mengidentifikasi gambar pada objek, YOLO juga harus berupaya penuh untuk dapat memisahkan dan melokasikan beberapa objek yang memiliki ukuran yang kecil.



Gambar 2. 9 Performasi kerja YOLO [24]

Berdasarkan Gambar 2.9 merupakan tingkatan performasi kerja dari YOLO dengan beberapa algoritma identifikasi objek lainnya. Pada gambar tersebut menunjukkan kelebihan YOLO yang memberikan performa terbaik dengan waktu pembelajaran dataset tersingkat dibandingkan algoritma lain. Dalam konteks YOLO, mengenal istilah *Darknet*. Dari kata *dark* menunjukkan bahwa lapisan yang dimiliki sangat banyak. *Darknet* merupakan arsitektur model *neural network* yang digunakan sebagai dasar untuk YOLO, dalam melakukan deteksi objek dari berbagai konteks, dengan model berupa *dataset* yang akan dilatih. Maka dari itu, istilah *darknet* lebih menuju kepada kerangka kerja atau *framework*. *Framework* yang dimaksud disini memiliki arti menyediakan dasar untuk membangun, melatih, dan mengevaluasi, mengakses, serta mengolah *dataset* yang nantinya diperlukan

untuk pelatihan. *Framework* juga mampu memberikan struktur serta panduan guna mempermudah pengembangan model. Maka dari itu, jika dihubungkan dengan YOLO sendiri diimplementasikan sebagai bagian algoritma deteksi objeknya.

2.2.10 PYTHON

Python merupakan bahasa pemrograman yang banyak digunakan bagi kalangan programmer untuk dapat menciptakan suatu aplikasi berbasis desktop, pengembangan dari perangkat lunak, hingga *Machine Learning* (ML). Bahasa pemrograman *Python* diciptakan oleh Guido Van Rossum pada tahun 1990. Bahasa pemrograman kemudian dikembangkan hingga sekarang banyak dipakai secara luas dalam implementasi teknologi karena memiliki struktur yang sederhana, ringkas, mudah untuk dipelajari, serta memiliki pustaka yang luas [25].

Manfaat yang didapat ketika menggunakan bahasa pemrograman *Python* yaitu sebagai berikut:

1. Python dapat diakses oleh berbagai sistem operasi komputer seperti *macOS*, *Windows*, *Linux*, dan *Unix*.
2. Terdapat banyak data pembelajaran yang dapat digunakan sebagai panduan dalam mempelajari bahasa pemrograman *python*.
3. Jika terdapat kerumitan dalam penggunaan bahasa pemrograman ini, dapat ditanyakan pada komunikasi *python* langsung karena memiliki jutaan *developer*.
4. *Programmer* dapat menggunakan *python* dengan bahasa pemrograman lain seperti *C++*, *C*, dan *Java*.
5. Bahasa pemrograman *python* memiliki pustaka yang besar yang berisikan kode yang dapat digunakan untuk menyelesaikan masalah, tanpa harus menggulung kode program dari awal.
6. *Python* menggunakan kode baris yang lebih sedikit dan lebih ringkas dibandingkan bahasa pemrograman lainnya.
7. Bahasa pemrograman *python* dapat dengan mudah dibaca dan dipahami karena memiliki sintaks dasar.

Fitur-fitur yang tersedia pada bahasa pemrograman *python* yaitu sebagai berikut:

1. Bahasa yang dapat diinterpretasikan

Bahasa pemrograman *python* merupakan bahasa yang dapat diinterpretasikan, dimana dalam menjalankan programnya *python* akan memulai secara runtut atau baris ke baris. Hal ini dapat membantu seorang *programmer* lebih mudah menemukan kode yang salah, karena *python* akan berhenti pada kode baris yang terdapat kesalahan dalam kode program.

2. Bahasa berorientasi objek

Bahasa pemrograman *python* akan melihat semua hal dalam bentuk objek, namun tidak menutup kemungkinan bahasa pemrograman ini mampu untuk melihat program yang tersusun terstruktur serta fungsional.

3. Bahasa pemrograman yang dapat dipahami manusia (*high level*)

Bahasa pemrograman *python* merupakan bahasa pemrograman yang sangat mudah untuk dimengerti oleh manusia dibandingkan dengan menggunakan bahasa pemrograman lainnya. Dimana bahasa pemrograman ini juga mendukung fungsionalitas program seperti arsitektur ataupun manajemen dari ruang penyimpanan.

4. Bahasa dapat diketik secara dinamis

Python dapat diaplikasikan dan dijalankan tanpa harus menyatakan tipe data dari tiap variabel data yang digunakan pada program, sehingga pengerjaan jauh lebih cepat.

Proses kerja dari bahasa pemrograman ini juga sudah difasilitasi dengan kumpulan modul serta paket. Seorang *programmer* dapat langsung menggunakan modul atau paket tersebut jika ingin membangun sebuah program dengan menggunakan *python*, sehingga proses pengerjaan dapat lebih cepat karena detail tingkat rendah pada *python* menjadikan program lebih cepat. *Python* memiliki dua jenis proses kerja yaitu sebagai berikut:

- Kerangka kerja tumpukan penuh, yang dimana digunakan ketika membuat program atau pembangunan aplikasi dengan cakupan yang besar.
- Kerangka kerja mikro, digunakan ketika ingin membangun aplikasi yang bersifat sederhana. Pada kerangka ini juga menyediakan ekstensi jika ingin membuat aplikasi menjadi lebih canggih dari versi sebelumnya.

2.2.11 CONFUSION MATRIX

Machine learning merupakan salah satu ilmu kecerdasan buatan yang dimana digunakan untuk membangun sistem berdasarkan data yang diberikan. Pada *macine learning* digunakan metode *Supervised Learning* yang merupakan model untuk memprediksi hasil yang akan didapatkan. Untuk dapat mengukur besarnya performa dari model pengelompokan atau klasifikasinya digunakanlah *Confusion Matrix*. *Confusion matrix* digunakan untuk menghitung nilai akurasi, presisi, *recall*, yang dimana metode evaluasi tersebut bermanfaat untuk mengukur performa dalam melakukan prediksi. *Confusion matrix* merupakan suatu tabel yang digunakan untuk menyatakan pengelompokan atau kalsifikasi dari jumlah data uji yang dinilai benar dan jumlah data uji yang dinilai salah [26].

Untuk hasil *output* yang diberikan dapat berjumlah dua kelas atau lebih yang terdiri dari nilai actual dan nilai prediksi. Nilai prediksi merupakan nilai yang belum tentu dapat dipastikan hasilnya atau masih dirancang, nilai prediksi ini memiliki kemampuan untuk memperbaiki kapasitas dalam pembuatan keputusan dalam masalah. Sedangkan nilai actual merupakan nilai yang terkini atau merupakan hasil didapat sebenarnya saat memecahkan suatu masalah, atau dapat dikatakan merukana nilai sesungguhnya yang didapat dari nilai prediksi. Di dalam confusion matrix dikenal dengan istilah 4 nilai prediksi dari hasil proses klasifikasi yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN).

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

Gambar 2. 10 Tabel *confusion matrix* [26]

Pada Gambar 2.10 merupakan ilustrasi dari tabel *confusion matrix* yaitu sebagai berikut:

1. *True Positive* (TP) merupakan nilai dengan jumlah data yang bernilai positif dan nilai prediksi yang ditunjukkan adalah benar sebagai positif. Contoh dalam deteksi bahasa isyarat alfabet BISINDO, maka TP dapat diartikan bahwa terdeteksi lambang tangan bahasa isyarat dan kenyataan memang benar bahasa isyarat alfabet BISINDO yang diperagakan.
2. *False Positive* (FP) merupakan nilai dengan jumlah data yang bernilai negatif tetapi nilai prediksi yang ditunjukkan adalah benar sebagai positif. Contoh dalam deteksi bahasa isyarat alfabet BISINDO, maka FP dapat diartikan bahwa terdeteksi lambang tangan bahasa isyarat namun kenyataannya tidak benar bahasa isyarat alfabet BISINDO yang diperagakan.
3. *False Negative* (FN) merupakan nilai dengan jumlah data yang bernilai positif tetapi nilai prediksi yang ditunjukkan adalah negatif. Contoh dalam deteksi bahasa isyarat alfabet BISINDO, maka FN dapat diartikan bahwa tidak adanya terdeteksi lambang tangan bahasa isyarat namun kenyataan memang benar bahasa isyarat alfabet BISINDO yang diperagakan.
4. *True Negative* (TN) merupakan nilai dengan jumlah data yang bernilai negatif dan nilai prediksi yang ditunjukkan adalah benar sebagai negatif. Contoh dalam deteksi bahasa isyarat alfabet BISINDO, maka TN dapat diartikan bahwa tidak terdeteksi lambang tangan bahasa isyarat dan kenyataan memang tidak benar adanya bahasa isyarat alfabet BISINDO yang diperagakan.

Adapun rumus untuk mencari nilai *accuracy*, *recall*, dan *precision* adalah:

1. *Accuracy*, digunakan untuk menggambarkan seberapa akurat model yang dibuat dalam mengklasifikasikan dengan benar dan tepat.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (2.1)$$

2. *Recall*, merupakan tingkatan yang menunjukkan seberapa banyak objek (*ground truth*) yang dapat dideteksi oleh sistem.

$$Recall = \frac{TP}{TP+FN} \quad (2.2)$$

3. *Precision*, digunakan untuk mengukur seberapa presisi hasil deteksi objek.

$$Precision = \frac{TP}{TP+FP} \quad (2.3)$$

2.2.12 VALIDASI DATA

Validasi data merupakan tahapan terpenting yang dilakukan saat ingin menguji model sistem yang telah dibuat apakah akan memberikan nilai prediksi yang cukup akurat atau tidak, sebelum akhirnya model tersebut siap untuk digunakan atau diproduksi [27]. Validasi model dapat digunakan untuk mengukur kinerja model yang dibangun dengan memperhitungkan setiap nilai akurasi, prediksi, serta nilai eror pada model. Nantinya nilai yang dihasilkan dapat dijadikan sebagai patokan parameter kinerja model itu sendiri. Dalam pengujian model kinerja sistem dapat dilakukan dengan 2 cara yaitu:

1. *Split Validation*

Split validation merupakan model pengujian sistem yang cukup sederhana dimana model akan dibagi menjadi dua bagian yaitu data *training* atau data latih serta data *testing* atau data yang nanti akan diuji ketika sistem belajar. *Split validation* ini juga dapat mengetahui model kinerja sistem berdasarkan nilai akurasi yang diukur. Terdapat 2 konsep pada *split validation* yaitu:

- *Training error*, validasi data yang didapatkan dari nilai atau letak dari kesalahan klasifikasi sistem terhadap data yang diberikan dengan data yang diuji pada model sistem.
- *Test error*, validasi data yang didapat dari kesalahan nilai klasifikasi sistem ada data yang diuji. Pada bagian ini setiap data yang ada pada sistem harus memiliki nilai label yang sama, dengan data yang dilatih dan diuji harus dalam data yang sepenuhnya terpisah. Pada bagian ini biasanya semakin banyak data yang dilatih maka akan menciptakan suatu model dengan sistem kinerja yang lebih baik dan tingkat akurasi yang lebih tinggi.

2. *Cross Validation*

Salah satu metode yang terdapat pada *cross validation* adalah *k-fold*. *K-fold cross validation* memiliki konsep kerja yang dimana sistem akan membuat beberapa model data uji atau data testing berulang kali, serta membagi *dataset*

menjadi data yang terpisah dengan ukuran tiap data yang sama. Nantinya model akan dilatih dengan menggunakan data *training* dan akan diukur validasinya dengan menggunakan data *testing* atau data uji. Model *k-fold cross validation* ini juga memiliki kelebihan dimana dapat mempersingkat waktu kerja sistem komputasi dengan tetap menjaga nilai akurasi dari model.

2.2.12 MEAN AVERAGE PRECISION (mAP)

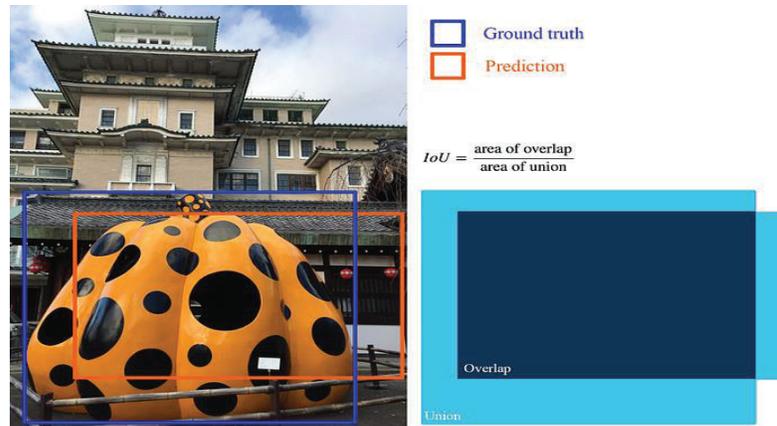
Mean Average Precision (mAP) adalah salah satu parameter yang digunakan untuk mengevaluasi serta mengukur kinerja sistem sebuah model deteksi objek seperti *Fast R-CNN*, *YOLO*, *Mask R-CNN*, dan lain-lain. mAP dapat dihitung dengan menghitung nilai rata-rata dari *Average Precision* (AP) [28]. mAP menggabungkan dua konsep penting dalam evaluasi pemulihan informasi, yaitu *precision* dan *recall*. Nilai dari mAP berkisar dari 0 sampai 100. Sebagai metrik evaluasi, semakin tinggi nilai mAP, semakin baik sistem deteksi objek tersebut dianggap dalam mengenali objek.

2.2.13 INTERSECTION OVER UNION (IoU)

Intersection over Union (IoU) merupakan *metric* yang umum digunakan dalam bidang visi komputer, terutama dalam menjalankan tugas pada deteksi objek dan juga segmentasi objek. *Intersection over Union* (IoU) adalah ratio atau ukuran kemiripan dari area yang tumpang tindih antara *bounding box* hasil prediksi dengan *ground truth bounding box* [29]. Dalam *Intersection over Union* (IoU) terdapat 2 kotak pembatas yaitu:

1. Kotak pembatas atau *Bounding Box Ground Truth*, merupakan kotak yang mengelilingi objek yang sebenarnya atau *ground truth* dalam tugas deteksi atau segmentasi objek. Kotak pembatas ini akan menandai area yang sebenarnya dan seharusnya diidentifikasi atau ditemukan sistem ketika melakukan deteksi objek.
2. *Bounding Box Prediction*, merupakan kotak yang mengelilingi objek yang ada pada saat melakukan deteksi objek. Seperti dari kata *prediction* yang berarti prediksi maka untuk kebenarannya bisa sepenuhnya benar, namun bisa juga tidak.

Dalam deteksi objek, seperti Gambar 2.11 menunjukkan bahwa *Intersection* mengacu pada area yang tumpang tindih antara *bounding box* prediksi dengan *bounding box ground truth* atau prediksi koordinat kotak pembatas dengan kotak kebenaran dasar. Sedangkan *Union* mengacu pada total area dari *bounding box* prediksi dan *bounding box ground truth*. Selanjutnya nilai *IoU score* diperoleh dengan membagi antara hasil *intersection* dengan *union*.



Gambar 2. 11 Intersection over union (IoU) [29]

Hasil yang didapatkan adalah bilangan antara 0 dan 1. Untuk nilai 1 menunjukkan tumpang tindih sempurna antar 2 area, sedangkan. nilai 0 menunjukkan tidak adanya tumpang tindih antara 2 area. Untuk nilai IoU yang lebih tinggi akan menunjukkan prediksi koordinat kotak pembatas sangat mirip dengan koordinat kotak kebenaran dasar sehingga akan menghasilkan indikasi tingkat akurasi yang lebih tinggi.