



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 COVID-19

Pada tahun 2020 terjadi bencana dengan skala global yaitu Pandemi COVID-19. COVID-19 sendiri ditemukan pertama kali pada bulan Desember tahun 2019 di Kota Wuhan, Provinsi Hubei, Negara China [7]. Tidak hanya berada di Wuhan, virus ini pun menyebar ke berbagai negara di seluruh dunia bahkan Indonesia. Kasus COVID-19 pertama di Indonesia ditemukan pada bulan Maret tahun 2020. Dalam waktu singkat virus ini pun menyebar secara cepat di Indonesia, yang dimana dalam waktu 1 bulan terdapat 134 kasus di berbagai daerah [8].

Pandemi ini memberikan dampak yang buruk secara luas, mulai dari rusaknya perekonomian hingga banyaknya korban yang meninggal dikarenakan terpapar oleh COVID-19. Berbagai cara dilakukan oleh pemerintah untuk melakukan pemutusan rantai penyebaran COVID-19, seperti PSBB maupun PPKM tidak hanya itu pemerintah juga gencar melakukan sosialisasi tentang pencegahan beserta penanganan COVID-19. Tidak hanya pemerintah yang berperan dalam hal tersebut, masyarakat juga berperan penting terutama dalam pelaksanaan sosialisasi penanganan COVID-19 yang disebut sebagai "Gerakan 5M" yaitu memakai masker, mencuci tangan, menjaga jarak, menghindari kerumunan, dan mengurangi mobilisasi.

Pada bulan Juni 2021 kasus COVID-19 sempat mengalami penurunan dan akhirnya pemerintah kembali memutuskan untuk menerapkan PPKM Darurat

pertama pada awal bulan Juli hingga tanggal 23 bulan Juli 2021 dan terus berlanjut hingga bulan Oktober 2021. Seiring menurunnya angka kasus COVID-19 dan dalam proses pemulihan ekonomi, beberapa perusahaan sudah mulai diizinkan kembali beroperasi dengan menjalani protokol kesehatan yang ketat.

2.2 Physical Distancing

Kementerian Kesehatan memberikan anjuran untuk melakukan *Social Distancing*, untuk meminimalisir penyebaran COVID-19. *Social distancing* adalah kegiatan jaga jarak minimal 1 meter dengan tujuan untuk mengurangi penyebaran COVID-19 [4]. Tapi seiring berjalannya waktu penerapan *social distancing* berubah menjadi *physical distancing*. *Physical distancing* sendiri sama dengan *social distancing*. Tetapi istilah tersebut diubah untuk membuat orang-orang tetap terhubung. Istilah *Physical Distancing* lebih cocok untuk digunakan dan lebih banyak disukai oleh banyak orang [9].

Physical Distancing merupakan metode yang tepat untuk diterapkan pada momen dimana banyak tempat terutama di dalam ruangan yang sudah kembali beroperasi. Tetapi metode ini punya kelemahan juga, memang metode ini merupakan salah satu yang mudah untuk dilakukan tetapi masih banyak orang yang kurang taat untuk melakukan *physical distancing*. Maka dari itu, sangat penting untuk dibuatnya sebuah sistem atau program yang nantinya berfungsi untuk memantau jarak *physical distancing* dan memberikan peringatan jika terjadinya pelanggaran terhadap salah satu protokol kesehatan tersebut.

2.3 Artificial Intelligence

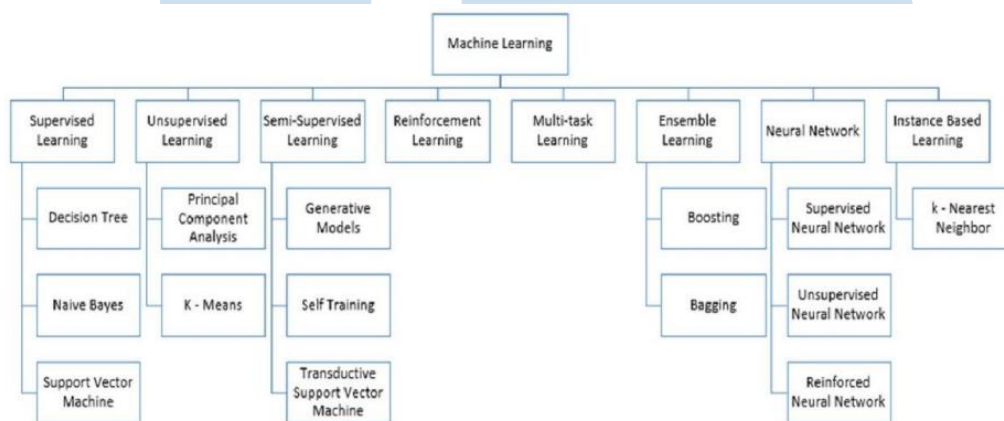
Artificial intelligence atau biasa disebut kecerdasan buatan adalah kecerdasan yang dimasukkan ke dalam suatu sistem sehingga sistem tersebut mampu mengartikan data eksternal serta mampu mengolah data tersebut dan memanfaatkan hasilnya untuk memberikan informasi. *Artificial intelligence* tidak selalu tentang robot, tetapi *artificial intelligence* juga mampu untuk memahami sifat pemikiran serta tindakan cerdas dalam menggunakan komputer sebagai perangkat uji coba. *Artificial intelligence* digunakan di berbagai sektor, seperti ekonomi, pendidikan, kesehatan, dan masih banyak lainnya [10].

Artificial intelligence di Indonesia juga dimanfaatkan pada berbagai sektor. Seperti yang paling banyak diketahui yaitu di industri manufaktur *artificial intelligence* tersebut banyak digunakan di bagian produksi. *artificial intelligence* dapat memberikan dampak positif maupun negatif. Dampak positif dari penerapan *artificial intelligence* yaitu mempersingkat waktu dan meningkatkan kualitas hasil proses. Pada beberapa negara berkembang *artificial intelligence* juga dapat memberikan dampak negatif dan penolakan dikarenakan *artificial intelligence* dapat mengurangi jumlah sumber daya manusia, serta memerlukan modal yang cukup besar.

2.4 Machine Learning

Machine learning merupakan bidang studi yang mengajarkan komputer untuk melakukan sebuah pengembangan terhadap hal yang dilakukan tanpa diprogram secara langsung. *Machine learning* sendiri memiliki fungsi untuk mengelola data dalam ukuran besar, sehingga data tersebut yang tadinya hanya sebuah kumpulan data biasa lalu dikelola hingga menjadi sebuah informasi. Dengan kemampuan

tersebut juga dimana jika semakin banyak data yang ada, maka semakin meningkat juga kemampuan komputer atau mesin untuk mengelola data tersebut dan menghasilkan informasi yang relevan. *Machine learning* memungkinkan untuk mengembangkan suatu model yang dilatih melalui rekaman data input maupun karakteristik luaran untuk meprediksi suatu kegagalan [6].



Gambar 2. 1 Algoritma *Machine Learning* [11]

Pada gambar 2.1 merupakan algoritma yang ada pada *machine learning*. *Machine learning* memiliki algoritma yang berbeda dalam memecahkan masalah. Jenis algoritma yang digunakan tergantung pada jenis masalah yang dihadapi, jumlah variabel, jenis model yang cocok, dan seterusnya [11].

2.5 Deep Learning

Deep learning merupakan cabang dari *machine learning* yang mencakup algoritma pemodelan yang mengabstraksi data tingkat tinggi menggunakan fungsi transformasi non-linear ditata secara mendalam dan berlapis. *Deep learning* berguna untuk beberapa kebutuhan seperti *supervised learning*, *unsupervised learning*, dan *semisupervised learning* pada berbagai aplikasi seperti pengenalan wajah, pengenalan suara, serta pengenalan pada teks [12].

2.6 Jupyter Notebook

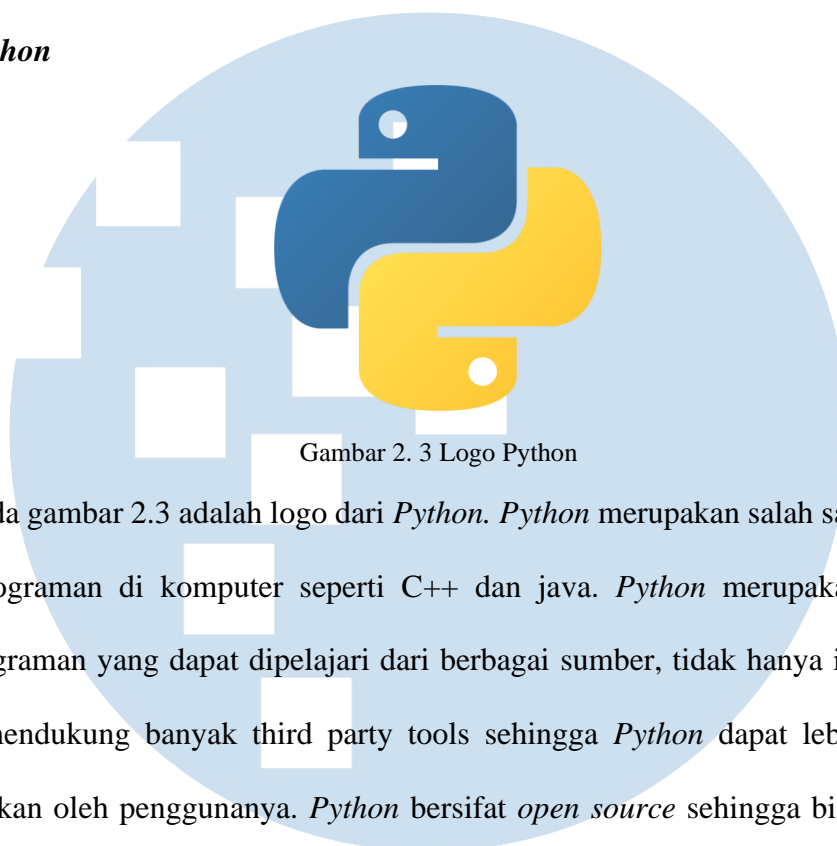


Gambar 2. 2 Logo Jupyter Notebook

Pada gambar 2.2 adalah logo dari *Jupyter*. *Jupyter* adalah organisasi non profit yang mengembangkan aplikasi interaktif dalam berbagai macam bahasa pemrograman. Salah satu aplikasi buatan *Jupyter* adalah *Jupyter Notebook*. *Jupyter Notebook* merupakan aplikasi berbasis web yang berfungsi untuk melakukan penulisan bahasa pemrograman python secara online, dapat digunakan untuk uji coba, serta berbagi hasil penulisan kepada orang lain. Alasan *python* dan *jupyter* semakin dikenal yaitu kecepatan dan kualitas aplikasi yang tinggi, banyaknya *library* seperti *SciPy*, *NumPy*, *Matplotlib*, dan lain lain. Tidak hanya itu *Jupyter Notebook* juga dapat terintegrasi dengan mudah dalam bahasa pemrograman C maupun C++. *Jupyter Notebook* juga bersifat gratis karena adanya dukungan dari komunitas besar yang terdiri dari *developer* dan pengguna [13].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.7 Python



Gambar 2. 3 Logo Python

Pada gambar 2.3 adalah logo dari *Python*. *Python* merupakan salah satu bahasa pemrograman di komputer seperti C++ dan java. *Python* merupakan bahasa pemrograman yang dapat dipelajari dari berbagai sumber, tidak hanya itu *Python* juga mendukung banyak third party tools sehingga *Python* dapat lebih mudah digunakan oleh penggunanya. *Python* bersifat *open source* sehingga bisa diakses oleh siapapun dan dapat dibagikan ke siapapun. Tidak seperti bahasa pemrograman C++, *Python* tidak terlalu memerlukan *memory management* yang rumit. Saat *Python* dijalankan bahasa tersebut secara otomatis langsung diubah ke bahasa yang dapat dimengerti oleh komputer [14].

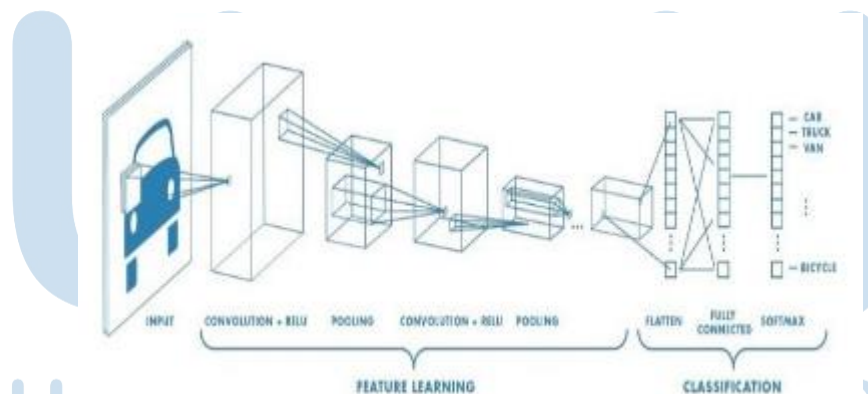
2.8 Computer Vision

Computer vision adalah kemampuan untuk menduplikasi kemampuan penglihatan pada manusia lalu diterapkan ke dalam benda elektronik, sehingga benda tersebut mampu memahami dan mengerti arti dari gambar yang dimasukkan. Data gambar dapat diartikan dalam berbagai bentuk, urutan video, sudut pandang dari beberapa kamera, ataupun data multi-dimensi. Contoh penerapan *computer vision* yaitu pengendalian proses seperti robot pada perusahaan manufaktur,

pengawasan secara visual seperti orang menghitung, mengorganisir informasi seperti mengurutkan gambar maupun melakukan pengindeksan pada foto, dan interaksi antara komputer maupun manusia [15].

2.9 *Faster R-CNN*

Faster R-CNN merupakan *convolution network* yang melakukan tugas untuk melakukan pendeteksian objek serta pembuatan *regional proposal network* atau proposal wilayah. Deteksi objek dapat dilakukan lebih cepat dengan menggunakan *Faster R-CNN*. Solusi dari masalah regional proposal diberikan oleh *Faster R-CNN* dengan penambahan RPN (*Regional Proposal Network*) sebagai komponen utama. Input pada pengumpulan ROI (*Region of Interest*) bukan gambar asli melainkan gambar yang sudah diolah dari *regional proposal*. *Faster R-CNN* merupakan model yang paling sedikit dalam melakukan komputasi dibandingkan dengan model *CNN* lainnya dikarenakan hasil pengolahan resolusi gambar dibuat lebih rendah daripada gambar aslinya [16].



Gambar 2. 4 Lapisan CNN[17]

Pada gambar 2.4 merupakan lapisan dari CNN. *Faster R-CNN* merupakan pengembangan dari CNN. CNN merupakan pengembangan dari *Multilayer*

Perception (MLP) yang berfungsi untuk mengolah data dalam bentuk dua dimensi. CNN termasuk ke dalam jenis *Deep Neural Network* karena terdapat kedalaman jaringan yang banyak serta tinggi sehingga mampu di aplikasi pada data citra khususnya pada klasifikasi citra [17].

2.10 YOLO



Gambar 2. 5 Tampilan Hasil YOLO[18]

Pada gambar 2. 5 merupakan tampilan dari YOLO. YOLO merupakan kepanjangan dari *You Only Look Once* adalah algoritma yang memiliki tehnik melakukan pendeteksian objek secara *realtime*. Pada gambar di atas menampilkan area yang ramai di suatu jalan. *Class* yang dapat dideteksi pada gambar 2.5 yang ditampilkan seperti lampu merah, orang, tas tangan, bis, truk, mobil, dan sebagainya. Algoritma YOLO dapat mengelompokkan dan mendeteksi setiap objek yang ada di gambar. Lalu pada objek yang terdeteksi akan menampilkan *bounding box*[18].



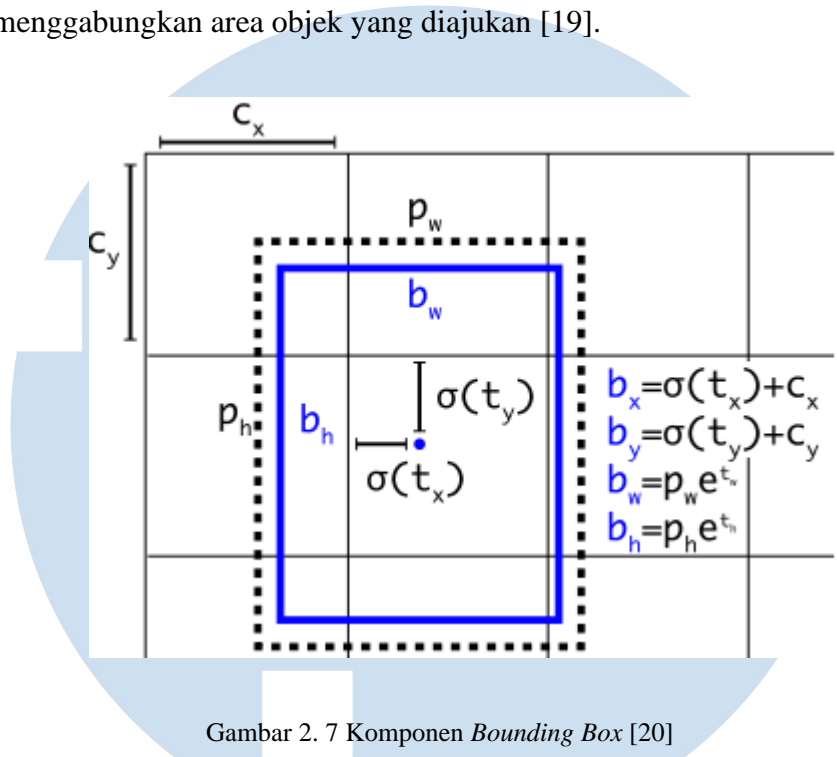
Gambar 2. 6 Tampilan *Grid Cell* [18]

Pada gambar 2.6 merupakan objek yang saling tumpang tindih. *YOLO* dapat mendeteksi objek lebih dari satu jika objek tersebut berada pada satu *grid cell*. *Grid cell* adalah kemampuan untuk melakukan pendeteksian objek pada suatu gambar. Objek yang terdeteksi dapat melakukan prediksi terhadap kelas dan lokasi pada objek di gambar tersebut. Pada satu *grid cell* dapat melakukan pendeteksian beberapa objek dengan menggunakan *anchor boxes* yang berisi titik pusat dari dua objek berbeda yang saling tumpang tindih. Pada gambar di atas menampilkan gambar yang objeknya saling tumpang tindih. Objek yang terdeteksi di gambar merupakan manusia dan mobil. *Anchor box* mencoba melakukan pendeteksian objek yang sesuai dengan menampilkan box dengan aspek rasionya[18].

2.11 *Bounding Box Regression*

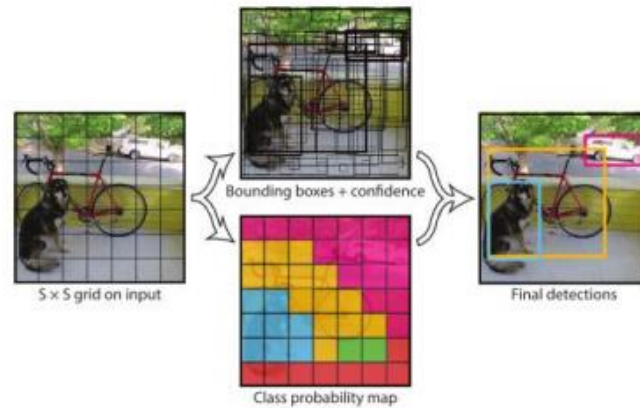
Bounding Box Regression merupakan teknik yang digunakan untuk memberikan tanda lokasi objek pada gambar dengan memberikan persegi pada objek yang terdeteksi. Seperti yang ditampilkan pada gambar 2.7 komponen yang ada di *bounding box* ada B_w dan B_h yang berfungsi sebagai titik tengah objek lalu

ada P_w dan P_h yang berfungsi sebagai jarak sisi objek. *Bounding box* berfungsi untuk menggabungkan area objek yang diajukan [19].



Setiap atribut pada *bounding box* akan diproses melalui normalisasi sehingga nilai yang dihasilkan adalah antara 0 dengan 1. Koordinat x dan h akan dinormalisasi sesuai dengan titik kiri atas pada *grid*. Tinggi dan lebar akan disesuaikan dengan ukuran gambar. Lalu nilai koordinat x dan h pada sebuah *bounding box* yang sudah dibagi menjadi *grid* akan menjadi titik tengah bagi *grid* tersebut [21]. Berikut pada gambar 2.8 merupakan bagaimana cara *YOLO* memproses suatu gambar untuk menghasilkan *bounding box*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 8 Bounding Box YOLO[21]

2.12 *Rapid Application Development*

Rapid application development merupakan strategi siklus hidup yang memiliki tujuan untuk menyediakan sistem atau aplikasi lebih cepat dan menyediakan hasil yang lebih baik daripada menggunakan siklus tradisional. *Rapid application development* merupakan kombinasi teknik terstruktur, teknik *prototyping*, dan teknik pengembangan *joint application* yang dapat mempersingkat waktu dalam pengembangan sistem dan aplikasi. Jika kebutuhan dan batasan ruang lingkup proyek diketahui, proses *Rapid application development* dapat membantu menciptakan sistem yang berfungsi penuh dalam waktu singkat [22].

Terdapat 4 tahap pada metode *rapid application development* yaitu *Requirement Planning*, *Design System*, *Rapid Construction*, dan *Cutover*. Pada tahap *Requirement Planning* merupakan tahap untuk melakukan identifikasi tujuan pada sistem yang akan dibuat berdasarkan pengumpulan informasi yang ada untuk mencapai pemecahan suatu masalah. Pada tahap *Design System* merupakan tahap untuk melakukan pembuatan desain dan menyesuaikan dengan kebutuhan saat diuji

coba. Pada tahap *rapid construction* dan *cuovver* merupakan tahap dari pengimplementasian sistem yang baru dibangun [23].

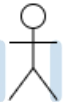

2.13 Unified Modeling Language

Unified modeling language merupakan metode pemodelan visual yang digunakan pada perancangan serta pembuatan pada aplikasi atau yang sistem yang berorientasikan pada objek. *Unified modeling language* merupakan standar penulisan yang didalamnya terdapat bisnis proses, serta penulisan kelas-kelas dengan penjelasan yang mendetail [24]. Terdapat beberapa bagian yaasng sering digunakan pada diagram *unified modeling language*, yaitu *use case*, *activity diagram*, dan *class diagram*.

2.13.1 Use Case

Use case adalah gambaran dari setiap fungsi yang ada pada sebuah sistem, lalu mempresentasikan interaksi antara sistem dengan aktor. Pada *use case* terdapat aktor, aktor merupakan gambaran manusia atau sistem yang menjalankan tugas pada sistem tersebut[24]. Berikut pada tabel 2.1 adalah simbol yang digunakan pada *use case*:

Tabel 2. 1 Simbol *Use Case*


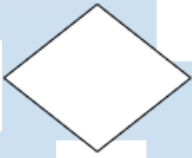
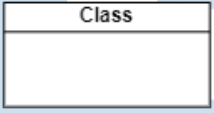
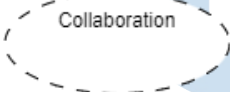
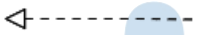


No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Menggambarkan manusia atau sistem yang menjalankan tugas pada sistem tersebut.
2		<i>Use Case</i>	Menampilkan urutan aksi pada sistem yang menghasilkan sesuatu yang terukur untuk aktor.

3		<i>Association</i>	Menghubungkan antara satu objek dengan objek lainnya.
4		<i>Dependency</i>	Hubungan yang terjadi perubahan pada elemen independent dan dapat mempengaruhi elemen lain yang bersifat dependent.
5		<i>Generalization</i>	Hubungan antara objek anak berbagi struktur data dan perilaku dari objek induk.
6		<i>Include</i>	Menjelaskan bahwa use case memiliki sumber eksplisit.
7		<i>Extend</i>	Menjelaskan bahwa use case target dapat memperluas perilaku dari use case sumber pada titik yang dituju.
8		<i>System</i>	Menjelaskan paket yang ditampilkan pada sistem secara terbatas.

2.13.2 Class Diagram

Class diagram bersifat statis, maka *class diagram* enunjukkan sekumpulan kelas, antarmuka, kolaborasi, dan hubungan. *Class diagram* didapat dari hasil analisa permasalahan terkait. Pada pembuatan *class diagram* akan membantu dan mempersingkat waktu dalam analisa masalah yang dihadapi[25]. Berikut pada tabel 2.2 adalah simbol yang digunakan pada *class diagram*:


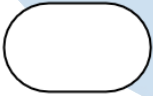



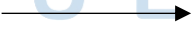
Tabel 2. 2 Simbol *Class Diagram*

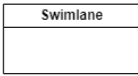
No	Simbol	Nama	Keterangan
1		<i>Generalization</i>	Hubungan antara objek anak berbagi struktur data dan perilaku dari objek induk.
2		<i>Nary Association</i>	Percobaan untuk menghindari hubungan yang lebih dari 2 objek.
3		<i>Class</i>	Himpunan setiap objek yang memiliki atribut serta operasi yang sama.
4		<i>Collaboration</i>	Menampilkan urutan aksi pada sistem yang menghasilkan sesuatu yang terukur untuk aktor. Menampilkan urutan aksi pada sistem yang menghasilkan sesuatu yang terukur untuk aktor.
5		<i>Realization</i>	Operasi yang dilakukan oleh sebuah objek.
6		<i>Dependency</i>	Hubungan yang terjadi perubahan pada elemen independent dan dapat mempengaruhi elemen lain yang bersifat dependent
7		<i>Association</i>	Penghubung antara satu objek dengan objek lainnya.

2.13.3 Activity Diagram

Activity diagram adalah gambaran alur dari aktivitas yang berjalan pada sistem. *Activity diagram* juga menjelaskan dari *workflow* pada proses sebuah sistem dan urutan aktifitas yang terjadi pada proses tersebut. *Activity diagram* mirip dengan *flowchart* karena memiliki model *workflow* dari suatu aktifitas menjadi status[26].

Tabel 2. 3 Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		<i>Activity</i>	Menampilkan proses setiap kelas saling berinteraksi satu sama lain.
2		<i>Action</i>	Kondisi pada sistem yang menggambarkan eksekusi suatu aksi.
3		<i>Initial Node</i>	Awal suatu objek dibentuk.
4		<i>Activity Final Node</i>	Penggambaran pengakhiran suatu objek.
5		<i>Decision</i>	Untuk menggambarkan keputusan atau tindakan yang diambil berdasarkan kondisi tertentu. Untuk menggambarkan keputusan atau tindakan yang diambil berdasarkan kondisi tertentu.
6		<i>Line Connector</i>	Untuk menghubungkan satu simbol dengan simbol lainnya.

7		<i>Swimlane</i>	Menampilkan pengganggu jawab dalam melakukan aktivitas pada suatu diagram.
---	---	-----------------	--

2.14 *User Acceptance Test*

User acceptance test adalah pengujian pada sistem yang telah dikembangkan dengan penguji yaitu *user* yang kemudian menghasilkan dokumen yang dapat digunakan sebagai bukti bahwa *user* telah menerima pengembangan aplikasi dan telah memenuhi kebutuhan pengguna berdasarkan tes yang telah dilakukan[27]. Setelah dilakukan uji coba sistem, melalui *user acceptance test* menyatakan bahwa sistem telah memenuhi persyaratan. *User acceptance test* biasanya menunjukkan bahwa sistem telah memenuhi persyaratan tertentu. *User acceptance test* mencakup lingkungan dan skenario yang sama pada saat uji coba yang berfokus pada penggunaan sistem yang dibuat [28].



2.15 Penelitian Terdahulu

Tabel 2. 4 Tabel Penelitian Terdahulu

No	Jurnal / Volume / Tahun	Judul Penelitian	Penulis	Hasil	Kesimpulan
1	<i>International Journal of Pervasive Computing and Communications</i> / Volume 17 / Tahun 2020	<i>Applying deep learning algorithm to maintain social distance in public place through drone technology</i>	Lalitha Ramadass, Sushanth Arunachalam, Sagayasree Z.	Sistem menggunakan <i>Python 3.6</i> dengan deep learning dan algoritma <i>YOLOv3</i> . Algoritma berfungsi untuk deteksi gambar dan video yang disebut <i>OpenCV</i> .	Algoritma bisa digunakan di kamera publik kemudian detailnya diambil ke unit kamera sistem detailnya sama dengan penggunaan drone lalu didsimpan di database.
2	<i>International Conference on Information Technology and Multimedia (ICIM)</i> / Volume 8 / Tahun 2020	<i>Social Distancing Detection with Deep Learning Model</i>	Yew Cheong Hou, Mohd Zafri Baharuddin, Salman Yussof, Sumayyah Dzulkifly	Algoritma <i>YOLO</i> mampu mendeteksi pejalan kaki sebagai objek. Untuk mengatasi kesalahan deteksi ditambahkan juga kotak persegi untuk mengamati wilayah tertentu.	Metode yang digunakan mampu untuk menentukan <i>social distancing</i> dan dapat digunakan di beberapa tempat seperti kantor, restoran, dan sekolah.
3	<i>Journal of Physics: Conference Series (JPCS)</i> / Volume 1916 / Tahun 2021	<i>Real-Time Social Distance maintaining using Image Processing and Deep Learning</i>	K R Senthil Murugan, G Kavindraj, K Mohanprasanth, Krishnan B Ragul	Resolusi video ditingkatkan agar dapat melakukan pengukuran jarak lebih lanjut. Masih terdapat kesalahan dalam pendeteksian	Orang akan terus dipantau dan bagi yang melanggar akan tampir garis merah. Sistem dapat ditingkatkan dengan menambah beberapa

				dikarenakan objek terlalu jauh antara satu dengan yang lain.	bantuan alat di jalan dan penyesuaian tampilan pada kamera.
4	Edu Komputika Journal / Volume 8 / Tahun 2021	Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19	Faizal Indaryanto, Anan Nugroho, Alfa Faridh Suni	Tingkat keberhasilan untuk mendeteksi manusia dan pelanggaran <i>social distancing</i> tinggi tetapi masih terdapat kegagalan karena beberapa hal seperti objek yang timpang tindih.	Orang akan terus dipantau dan bagi yang melanggar akan tampir garis merah. Sistem dapat ditingkatkan dengan menambah beberapa bantuan alat di jalan dan penyesuaian tampilan pada kamera.
5	PLoS ONE Journal / Volume 15(11) / Tahun 2020	<i>Automated measurement of anteroposterior diameter and foraminal widths in MRI images for lumbar spinal stenosis diagnosis</i>	Friska Natalia, Hira Meidia, Nurnik Afriliana, Julio Christian Young, Reyhan Eddy Yunus, Mohammed Al-Jumaily, Ala Al-Kafri, Sud Sudirman	Metode yang digunakan ketika algoritma diterapkan dapat bekerja dengan baik untuk meningkatkan akurasi gambar label yang digunakan untuk melatih model SegNet dan gambar yang sudah disegmentasi secara otomatis.	Kinerja algoritma yang disarankan untuk dievaluasi melalui percobaan pada dataset MRI Lumbar Spine tersedia untuk umum yang berisi pembahasan MRI dari 515 pasien dengan nyeri punggung simtomatik.

2.16 Perbandingan Penelitian Terdahulu

Pada beberapa penelitian terdahulu menggunakan algoritma yang sama yaitu *YOLOv3* tetapi hasil yang diberikan pada setiap akhir penelitian berbeda. Mulai dari penjelasan algoritma dan penggunaan *library* apa yang digunakan seperti yang dijelaskan pada jurnal [29], lalu solusi jika objek manusia tidak terdeteksi maka diharuskan menambahkan kotak untuk mengamati wilayah tertentu pada jurnal [30], lalu pada jurnal [31] juga menjelaskan untuk dapat melakukan pengukuran jarak yang lebih jauh bisa dengan cara meningkatkan resolusi video, dan pada jurnal [32] juga melakukan pengukuran seberapa besar untuk menampilkan tingkat keberhasilan algoritma tersebut terhadap objek manusia dan *social distancing*. Lalu penelitian [33] merupakan basis awal dari penelitian ini.

