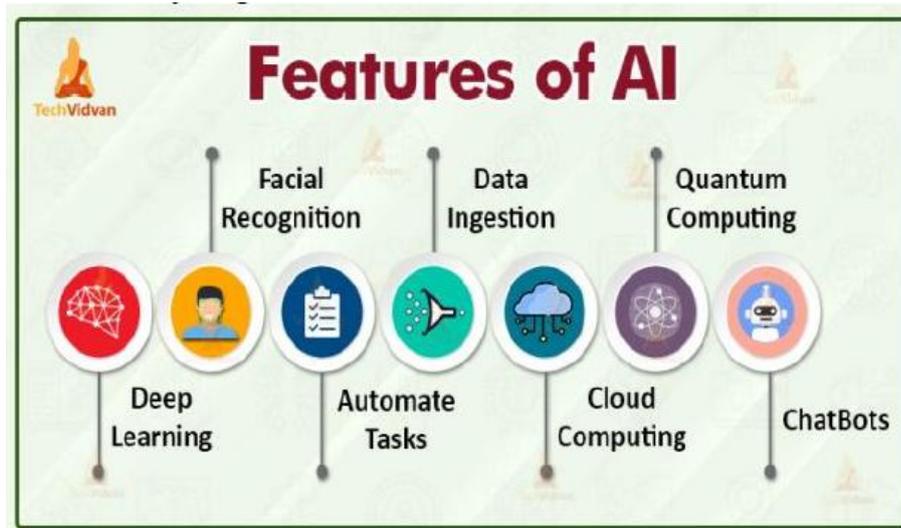


BAB II

TINJAUAN PUSTAKA

2.1. *Artificial Intelligence*

Perkembangan teknologi yang semakin pesat di dunia ini menyadarkan banyak orang bahwa saat ini dunia sedang berada pada posisi keempat pada revolusi industri yang di mana teknologi mengaburkan batasan antara bidang fisik, digital, dan biologis [21]. Salah satu teknologi yang sangat berkembang dan berpengaruh pada era saat ini adalah kecerdasan buatan atau *Artificial Intelligent (AI)* [22]. Ai menjadi penggerak revolusi industri 4.0 karena menyediakan banyak kemudahan baik kepada pemerintah maupun industri [23]. Teknologi kecerdasan buatan ini telah menjangar ke berbagai bidang, salah satunya adalah Sistem Informasi [22], [23], [24]. Pemaparan dari fitur-fitur kecerdasan buatan dapat dilihat pada Gambar 2.1.

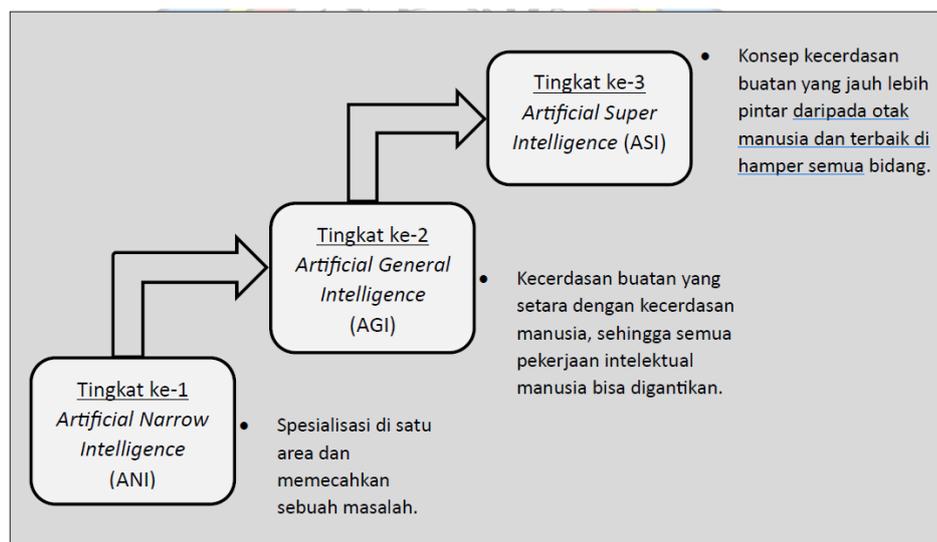


Gambar 2. 1 Tugas-tugas Kecerdasan Buatan

Berdasarkan Gambar 2.1, diketahui bahwa *Deep Learning*, *Facial Recognition*, otomatisasi, penyerapan data, *Cloud Computing*, *Quantum Computing*, dan *Chatbot* adalah contoh-contoh dari fitur yang diberikan oleh kecerdasan buatan. Kecerdasan buatan suatu sistem pada komputer yang dapat menyelesaikan tugas-tugas yang biasanya dibutuhkan kemampuan kecerdasan manusia. Proses yang terdapat pada AI mencakup *learning*, *reasoning*, dan *self-correction*, yang di mana hal tersebut mirip seperti manusia berpikir sebelum mengambil keputusan [21], [25].

Berdasarkan Luger dan Wiliam, kecerdasan buatan adalah suatu cabang dari ilmu komputer yang berhubungan pada sistem otomatisasi perilaku yang cerdas. Sedangkan, menurut Haag dan Peter, kecerdasan buatan adalah suatu bidang studi yang berkaitan dengan penangkapan, pemodelan, dan penyimpanan kecerdasan manusia ke dalam suatu sistem teknologi informasi sehingga sistem tersebut dapat mengambil keputusan-keputusan selayaknya manusia [26]. Dengan kecerdasan buatan, komputer dapat belajar dan beradaptasi dari data yang telah diberikan yang berguna untuk mengambil keputusan secara sendirinya tanpa adanya bantuan atau intervensi dari manusia [22].

Kecerdasan buatan memiliki 3 tingkatan yang berbeda, ketiga tingkatan tersebut dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Tingkatan Kecerdasan Buatan

Berdasarkan Gambar 2.2 diketahui bahwa 3 tingkatan pada kecerdasan buatan, yakni ANI, AGI dan ASI. Berikut ini adalah penjelasan dari masing-masing tingkatan kecerdasan buatan.

- Tingkatan pertama, *Artificial Narrow Intelligence (ANI)*, pada tingkatan ini AI hanya mampu memecahkan suatu permasalahan saja, sehingga sering disebut kecerdasan buatan yang sempit (*Narrow AI*).
- Tingkatan kedua, *Artificial General Intellegence (AGI)*, merupakan kecerdasan buatan ini setara dengan tingkat kecerdasan manusia dan mampu memecahkan berbagai permasalahan yang sebelumnya harus manusia lakukan bisa

digantikan dengan sistem atau robot. Oleh sebab itu, kecerdasan buatan ini sering juga disebut sebagai *Human Level Artificial Intelligence* (HLAI).

- c) Tingkatan ketiga adalah *Artificial Super Intelligence* (ASI), merupakan kecerdasan buatan yang jauh lebih pintar daripada otak manusia. yang membedakan antara kecerdasan buatan ASI dengan kecerdasan manusia terdapat pada jumlah neuron, jika pada manusia terbatas pada beberapa miliar neuron, pada ASI jumlah neuron tidak terbatas [27].

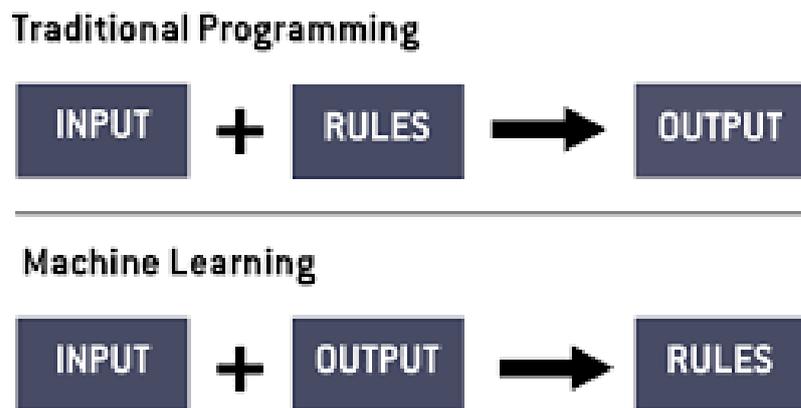
Kecerdasan Buatan telah memasuki berbagai bidang kehidupan, salah satunya adalah pendidikan dan sains [28][29]. Pemanfaatan aplikasi berbasis kecerdasan buatan, seperti ChatGPT, yang merupakan suatu jenis *chatbot* berbasis kecerdasan buatan yang dapat mengetahui *input* dan menghasilkan *output* berupa bahasa naturalnya manusia [30]. Selain keunggulan yang dapat diambil dari ChatGPT dalam proses pembelajaran, terdapat juga dampak negatif yang dihasilkannya, sehingga diperlukan tanggung jawab dan bijaksana dalam memakai aplikasi ini.

2.2. *Machine Learning*

Pembelajaran Mesin atau *Machine Learning* (ML) merupakan suatu metode dalam ilmu komputer yang memungkinkan komputer dapat mempelajari sesuatu berdasarkan data yang diberikan untuk menemukan pola dan menghasilkan keputusan yang diharapkan [31]. *Machine Learning* memiliki kemampuan yang meniru manusia dalam proses pembelajaran manusia dan mampu mengetahui pemahaman yang kompleks sehingga meningkatkan efisiensi komputer [32]. *Machine Learning* dapat melakukan tugas-tugas tertentu berdasarkan data yang diberikan kepada sistem dengan mempelajari algoritma dan model statistik yang ada [33], [34]. *Machine Learning* menjadi suatu fondasi dari banyaknya aplikasi yang penting, seperti *web search*, pengenalan suara, rekomendasi produk, email anti-spam, dan lain sebagainya [35].

Kemampuan *Machine Learning* ini menjadikannya lebih unggul jika dibandingkan dengan pemrograman tradisional. Jika pemrograman tradisional membuat aplikasi berdasarkan data *input* dan serangkaian program, berbeda dengan *Machine Learning* yang membuat suatu program berdasarkan data *input* dan *output*-nya untuk dipelajari oleh mesin. Data yang diberikan kepada mesin memiliki

fitur dan label, setiap fitur-fitur akan dipelajari untuk menghasilkan labelnya untuk setiap fitur. Oleh sebab itu, penggunaan *Machine Learning* sangat unggul untuk dipakai jikalau permasalahannya terlalu kompleks bila dilakukan dengan cara pemrograman tradisional. Perbedaan antara pemrograman tradisional dengan *Machine Learning* dapat diilustrasikan pada Gambar 2.3.



Gambar 2. 3 Diagram Perbandingan Pemrograman Tradisional vs *Machine Learning*

Berdasarkan Gambar 2.3, diketahui bahwa pemrograman tradisional membutuhkan peraturan-peraturan, yakni berupa program, agar program tersebut dapat menghasilkan keluaran yang diharapkan berdasarkan *input* yang diberikan. Sedangkan pada konsep *machine learning*, diketahui bahwa program tidak membutuhkan peraturan-peraturan secara spesifik oleh *programmer*, melainkan menaruh *input* dan *output* yang diharapkan pada suatu program, kemudian program tersebut dengan kemampuan pembelajarannya akan membentuk *rules* sendiri yang sungguh kompleks. Sehingga, pada konsep *machine learning*, program akan dapat secara otomatis menghasilkan jawaban berdasarkan seluruh *input* yang sudah dipelajari oleh mesin sebelumnya. Dengan mengambil manfaat dari AI dan *Machine Learning*, banyak aktivitas yang sukar untuk diselesaikan hanya dari sisi *engineering* menjadi lebih mudah untuk diselesaikan [23], [36]. Bidang *Machine Learning* terpopuler untuk saat ini adalah *Predictive Modelling* yang mampu memprediksi hasil berdasarkan tugas yang diberikan, sistem ini sudah dipakai banyak perusahaan besar, seperti Amazon, Alibaba, dan perusahaan raksasa lainnya [36]. Selain itu, pemanfaatan model *Machine Learning* juga dapat diterapkan dalam industri kesehatan seperti pendeteksi dini penyakit kardiovaskuler [37]. Juga dalam

industri pasar modal, model *machine learning* dapat dipakai untuk mendeteksi harga saham pada waktu yang akan mendatang [38].

Meskipun banyak kecanggihan yang ditawarkan oleh *Machine Learning*, keamanan dan perlindungan dari sistem juga harus diperhatikan. Hal ini karena adanya serangan dari pihak ketiga yang dapat melawan model ML dan menjadikan model salah dalam mengklasifikasikan *input*, mengurangi akurasi prediksi, dan mengekstraksi data *training* model, bahkan mencuri datanya [31], [39]. Sehingga dibutuhkan suatu studi untuk mencegah ini, yakni *Adversarial Machine Learning* (AML) yang berfokus pada eksploitasi kerentanan ML dan mengembangkan mekanisme pertahanan untuk mitigasi eksploitasi kerentanan ML [31], [40]. Oleh sebab itu, dibutuhkan kebijaksanaan dan tanggung jawab dalam menggunakan ataupun membuat sistem *Machine Learning* agar data privasi yang bersifat rahasia tidak terekspos dan dijadikan hal negatif yang berdampak merugikan.

2.2.1. *Deep Learning*

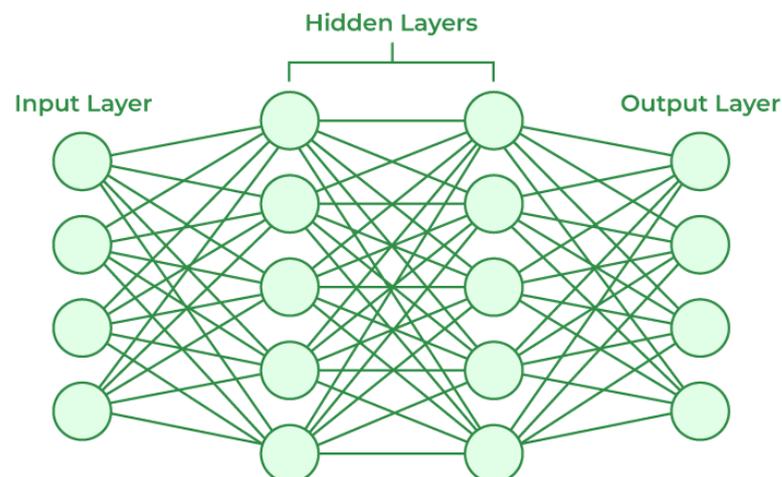
Deep Learning adalah salah satu cabang *Machine Learning* yang menjadi suatu solusi dalam pembelajaran hierarki (*Hierarchical Learning*) yang dapat membuat model mengabstraksikan data masukan secara bertingkat [41]. Model yang menggunakan konsep *deep learning* mampu secara otomatis mempelajari data dan meningkatkan fungsinya dengan menganalisis algoritma. *Deep Learning* sangat efektif karena mampu mempelajari data tak terstruktur (*unstructured data*), seperti gambar, video, audio, bahasa natural manusia, bahkan dataset medis seperti CT-Scan dan MRI [42]. Beberapa contoh dari lapisan *deep learning* adalah *dense layer* untuk menemukan pola data, *convolutional layer* untuk mengekstrak fitur-fitur pada gambar, *recurrent layer* untuk pemrosesan bahasa natural ataupun data kontinu, dan *pooling layer* untuk mereduksi parameter model.

Model *deep learning* menghasilkan akurasi yang lebih besar daripada pembuatan model dengan mengekstrak fitur secara manual, serta dengan menggabungkan beberapa model *deep learning* dengan *dense layer* akan menghasilkan akurasi klasifikasi yang tinggi [43]. Penelitian terkini, seperti dibentuknya dataset besar dan GPU, menjadikan *deep learning* sebagai solusi yang terjangkau untuk kendala *hardware* [41]. Karena *deep learning* saat ini dapat

dengan mudah dibentuk melalui berbagai *framework* seperti TensorFlow dan PyTorch. Selain itu *deep learning* juga dapat dibentuk dengan menggunakan layanan *notebook cloud*, seperti Jupyter Notebook, Google Colaboratory, dan Kaggle Notebook

2.2.2. Artificial Neural Network

Artificial Neural Network merupakan suatu model *Machine Learning* yang terinspirasi dari jaringan saraf biologis pada makhluk hidup [44]. ANN merupakan inti dari *Deep Learning*, karena kemampuannya yang mampu menyelesaikan tugas kompleks ML, seperti klasifikasi miliaran gambar, penguatan layanan pengenalan suara, dan rekomendasi video. Tugas umum yang dilakukan dengan menggunakan ANN adalah regresi dan klasifikasi [44]. Model pertama sekali dari ANN ditemukan oleh ahli saraf Warren McCulloch dan matematikawan Walter Pitts pada tahun 1943, yang dalam penelitiannya mereka menyimplifikasikan model komputasional dari cara kerja jaringan saraf pada otak binatang untuk melakukan komputasi yang kompleks menggunakan *propositional logic* [44]. Ilustrasi dari model jaringan saraf buatan atau *Artificial Neural Network* (ANN) dapat dilihat seperti pada Gambar 2.4.



Gambar 2. 4 Diagram *Artificial Neural Network* (ANN)

Berdasarkan Gambar 2.4, didapati bahwa terdapat 3 lapisan, yakni *input layer*, *output layer*, dan *hidden layer*. *Input layer* merupakan layer masukan, sedangkan *output layer* merupakan layer keluaran. Sedangkan, *hidden layer*

merupakan sejumlah lapisan dengan algoritma-algoritma tertentu yang direkayasa berdasarkan kebutuhan dari program. Kemudian model ANN semakin berkembang dan ditemukannya *Perceptron*, *Perceptron* merupakan model paling sederhana dari ANN yang ditemukan oleh Frank Rosenblatt pada tahun 1957.

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n \quad (2.1)$$

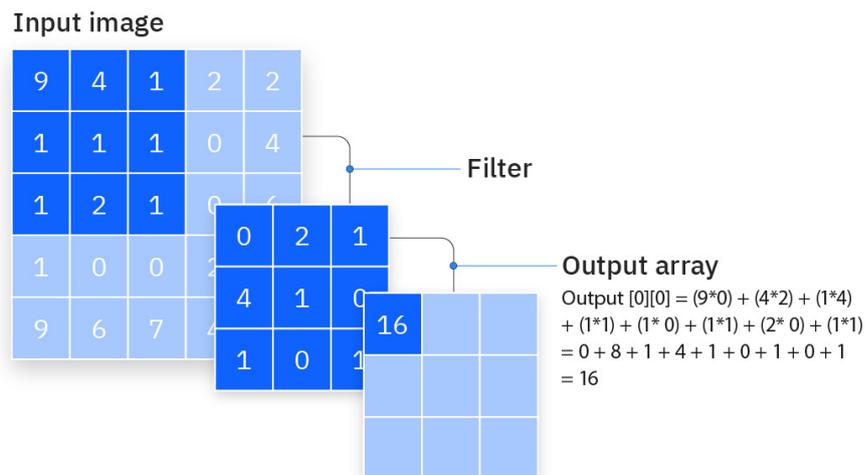
$$h_w(x) = \text{step}(z) \quad (2.2)$$

$$\text{Heaviside}(z) \begin{cases} 0 & \text{jika } z < 0 \\ 1 & \text{jika } z \geq 0 \end{cases} \quad \text{sgn}(z) \begin{cases} -1 & \text{jika } z < 0 \\ 0 & \text{jika } z = 0 \\ 1 & \text{jika } z > 0 \end{cases} \quad (2.3)$$

Semua masukan dan keluaran dari *perceptron* adalah angka, dan setiap koneksi masukan terasosiasi dengan suatu beban (*weight*), kemudian akan dijumlahkan seluruh *input* seperti pada persamaan 2.1. Lalu dilakukan *step function* untuk hasil penjumlahan tersebut dan hasil keluarannya hasil fungsi *step* dari z seperti pada persamaan 2.2 [44]. Umumnya, fungsi *step* yang ada pada *perceptron* adalah *Heaviside step function* dan *Sign function* yang dapat dilihat pada persamaan 2.3.

2.2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan bagian dari *deep learning*, terdiri dari beberapa lapisan simpul. CNN memiliki suatu *input* dan *output*, dan diantaranya terdapat satu atau beberapa lapisan, yang disebut sebagai *hidden layer*. Lapisan konvolusional terdiri dari beberapa komponen atau *hyperparameter*, seperti *filter size*, *stride*, dan *padding*. Filter pada CNN berguna sebagai *feature detector* yang mendeteksi setiap *pixel* pada gambar untuk mendapatkan fitur-fitur pada gambar, proses tersebut dinamakan proses konvolusi. Selama filter mendeteksi *pixel-per-pixel* pada gambar, secara bersamaan dilakukan *dot product* sebagai hasil keluarannya. Ilustrasi dari algoritma *Convolutional Neural Network* (CNN) dapat dilihat pada Gambar 2.5.



Gambar 2. 5 Diagram *Convolutional Neural Network* (CNN)

Berdasarkan Gambar 2.5, didapati suatu contoh *input* gambar dengan nilai matriks 5x5 dengan angka bulat, Lalu terdapat matriks kedua yang berukuran 3x3 yang merupakan *filter* untuk *input* gambar. Hasil keluaran gambar merupakan hasil *Dot Product* antara matriks pada *input* gambar dengan matriks filter. Ukuran filter umumnya adalah matriks 3 x 3 dengan nilai-nilai khusus setiap elemen pada matriksnya. Filter akan bergerak pada suatu *stride*, dan mengulanginya sampai kepada titik akhir pada gambar dan menghasilkan *output* berupa *feature map*, *activation map*, atau *convolved feature*. Metode CNN secara meluas dimanfaatkan untuk klasifikasi gambar karena mampu menghasilkan performa yang baik pada prediksi, *staging*, dan prognosis dari CRC [42], [45], [46]. Sebagai *classifier*, biasanya model terbentuk dengan lapisan-lapisan konvolusi (*convolutional layers*) dan diakhiri dengan lapisan linier (linier layer). Lapisan konvolusi diawali ini digunakan untuk mengekstraksi fitur-fitur data yang kompleks, kemudian diakhiri dengan reduksi dimensi dan klasifikasi menggunakan lapisan linier [47].

2.3. Computer Vision

Visi Komputer (*Computer Vision*) merupakan cabang dari *Machine Learning* yang memungkinkan mesin untuk dapat memahami dan menginterpretasi informasi visual, dan berdasarkan informasi tersebut mesin dapat mengidentifikasi, melacak, dan mengklasifikasi. Misalnya dapat melacak dan mengklasifikasi kecacatan produk dalam industri manufaktur dengan menganalisis gambar atau video [48].

Computer Vision dipakai pada sistem klasifikasi kanker kulit menggunakan model *deep learning*, seperti CNN dan model *pre-trained deep learning* [49]. *Computer Vision* pernah dipakai untuk melawan dan mengontrol pandemik COVID-19, seperti penggunaan *Computed Thermography (CT)*, *X-ray Imagery*, *Ultrasound Imaging*, dan pencegahan serta pengontrolan [50]. *Computer vision* dipakai untuk meningkatkan mekanisme *quality control* menggunakan 3 jenis *Machine Learning*, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning* [51]. Beberapa contoh tugas dalam *Computer Vision* adalah *Image Classification* dan *Object Detection* seperti yang dijelaskan di bawah berikut ini

2.3.1. *Image Classification*

Image Classification merupakan tugas dalam *Computer Vision* yang membuat model dapat mengklasifikasikan kelas pada gambar yang diberikan. Misalnya, model yang diberikan gambar kucing akan memberikan hasil keluaran kucing, seperti yang dapat dilihat pada Gambar 2.6.



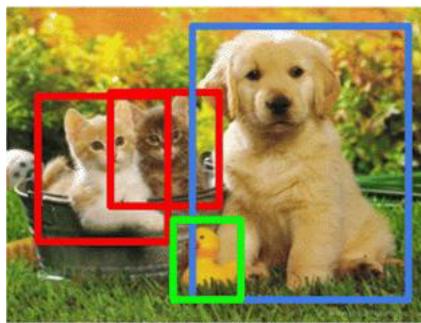
CAT

Gambar 2. 6 Contoh Klasifikasi Gambar

Berdasarkan Gambar 2.6, suatu model dapat mengklasifikasikan bahwa gambar yang diberikan adalah gambar seekor kucing berdasarkan sistem *Machine Learning* yang dilakukan *training* dengan menggunakan gambar-gambar kucing. Model yang umum dalam *Computer Vision* untuk tugas klasifikasi gambar adalah ResNet dan VGGNet. ResNet adalah suatu model yang memanfaatkan arsitektur CNN dan *dense layer* untuk menemukan pola kompleks pada informasi visual. Sedangkan VGGNet adalah model yang berbasis arsitektur CNN dengan memakai filter dengan matriks ukuran 3 x 3.

2.3.2. Object Detection

Berbeda dengan klasifikasi gambar, tugas deteksi objek pada visi komputer adalah dengan melacak posisi keberadaan suatu objek dengan kelas tertentu. Tugas deteksi objek memungkinkan model untuk mengklasifikasikan lebih dari satu kelas dan memberikan *bounding-box* pada kelas-kelas yang terdeteksi dalam satu gambar. Misalnya, terdapat suatu gambar yang diberikan kepada model, di mana gambar tersebut terdapat 3 kelas objek, yakni anjing, kucing, dan angsa. Model tersebut akan dapat mengklasifikasikan seluruhnya serta memberi *bounding-box* pada lokasi setiap objek yang terdeteksi, seperti yang dapat dilihat pada Gambar 2.7.



CAT, DOG, DUCK

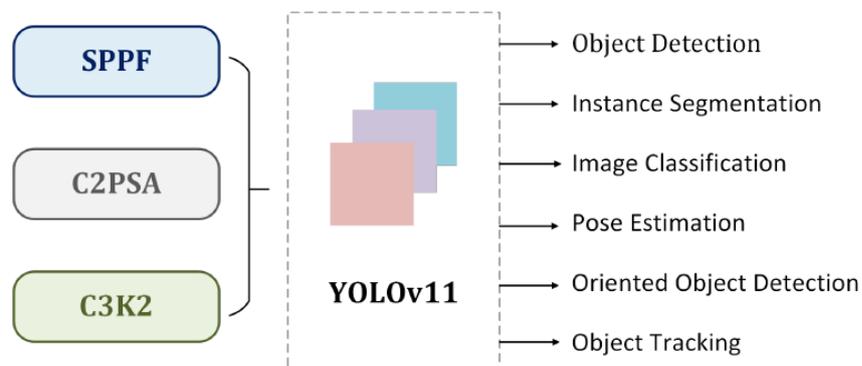
Gambar 2. 7 Contoh Deteksi Objek

Berdasarkan Gambar 2.7, model dapat mendeteksi objek dengan cara mengklasifikasikan objek dan memberikan hasil keluaran berupa *bounding box* pada setiap objek yang berhasil diklasifikasi. Penugasan Deteksi objek berbeda dengan klasifikasi, seperti yang dapat dilihat pada Gambar 2.6, keluaran model tidak memprediksi *bounding box* pada objek yang terdeteksi. Terdapat beberapa model yang diciptakan untuk penugasan untuk deteksi objek, yang berfungsi untuk melakukan klasifikasi dan regresi *bounding box*. Model-model yang umum digunakan pada tugas deteksi objek adalah *Faster R-CNN*, *YOLO*, dan *SSD*.

2.4. Algoritma *You Only Look Once*

YOLO merupakan singkatan dari *You Only Look Once* yang adalah suatu model yang menggunakan *convolutional neural network* untuk mendeteksi kelas objek dan *bounding-box* secara bersamaan dari suatu gambar. Model YOLO mengalami perkembangan dari seiring perkembangan teknologi, untuk saat ini terdapat 11 versi model YOLO, dari YOLOv1 hingga YOLOv11. YOLO menjadi suatu model pendeteksi yang bersifat objek *one-stage*, diikuti dengan SSD dan RetinaNet. Hampir dari seluruh seri dari model YOLO dilatih dengan menggunakan dataset Ms COCO [52]. Dataset Ms COCO atau COCO adalah suatu dataset dari Microsoft yang merupakan sekumpulan gambar dalam jumlah yang sangat besar yang digunakan secara umum untuk *object detection*, *captioning*, *segmentation*, dan lain sebagainya.

Seri terbaru dari algoritma YOLO saat ini adalah YOLOv11. YOLO versi ke-11 ini dibentuk dengan pendalaman yang signifikan sebagai teknologi real-time *object detection* dari seri-seri sebelumnya [53]. Gambar ringkasan arsitektur dari YOLOv11 dengan penugasan-penugasan yang dapat dilakukannya dapat dilihat seperti yang tertera pada Gambar 2.8.



Gambar 2. 8 Arsitektur Algoritma YOLOv11

Berdasarkan Gambar 2.8, arsitektur YOLOv11 terdiri dari 3 komponen yakni *Backbone*, *Neck*, dan *Head* [54]. Komponen *Backbone* menjadi komponen utama ekstraksi fitur dan memanfaatkan CNN untuk mengubah data gambar mentah menjadi pemetaan fitur multi-skala (*multi-scale feature maps*). Komponen berikutnya, yakni *Neck*, bekerja sebagai perantara pada tahap pemrosesan, serta menggunakan lapisan-lapisan terspesialisasi (*specialized layers*) untuk melakukan

agregasi dan meningkatkan representasi fitur-fitur pada berbagai skala yang berbeda. Dan terakhir adalah komponen *Head*, bertugas dengan memiliki fungsi sebagai mekanisme prediksi dan melahirkan hasil keluaran final (*final output*) serta klasifikasi pemetaan fitur yang diperhalus (*refined feature maps*). Lapisan *backbone* pada YOLOv11 terdiri dari lapisan konvolusi, SPPF dan C2PSA. Kemudian, pada komponen *neck* terdiri atas blok C3K2 dan *Attention Mechanism*. Lalu pada komponen akhir, *head*, terdiri atas blok C3K2, blok CBS, lapisan-lapisan konvolusi final, dan lapisan deteksi.

2.5. Kajian Pustaka

Pada referensi [13], metode yang digunakan adalah algoritma YOLOv3 sebagai deteksi orang dan algoritma *Simple Online and Real-Time Tracking* (SORT) sebagai pelacak yang membandingkan objek yang terdeteksi pada *frame* sebelumnya, untuk memastikan bahwa perubahan *frame* tidak mengubah *id* dari objek yang terdeteksi di *frame* sebelumnya. Membuat sistem pendeteksi dan pelacak orang secara *real-time* dengan algoritma YOLO. Dari hasil eksperimen yang membandingkan model YOLOv3 dengan Faster R-CNN, YOLO, dan SSD512, model YOLOv3 mendapatkan performa yang jauh lebih unggul dari metode lainnya, yakni mendapatkan nilai mAP sebesar 89.69% dan FPS sebesar 58.31.

Pada referensi [14], metode yang digunakan pada penelitian ini adalah Kalman filter sebagai pelacak dan LOI (*Line of Interest*) sebagai penghitung jumlah pengunjungnya. Metode yang digunakan diterapkan pada *microcomputer* Raspberry Pi yang secara meluas telah digunakan pada sistem-sistem berbasis *real-time image processing*. Membuat sistem yang murah dan mudah digunakan untuk mendapatkan pola pengunjung pada toko retail kecil yang dapat bekerja pada transmisi nirkabel untuk membawakan informasi statistik yang akurat mengenai jumlah orang yang ada pada toko. Penelitian ini membandingkan yang dibentuk dengan metode *Flow Mosaicking* (FM), *Directional People Counter* (DPC), *Counting People Crossing a line* (CPC) yang diuji pada 4 dataset, yakni SU1, SU2, DU1, dan DU2. Secara keseluruhan, metode yang digunakan menghasilkan *error* sebesar 45, jauh lebih kecil dari pada CPC dengan *error* sebesar 83, DPC dengan *error* sebesar 325, dan FM dengan *error* sebesar 154. Dengan berdasarkan performa

rata-rata *Recall* sebesar 0.98, rata-rata *Precision* sebesar 0.96, rata-rata F1 sebesar 0.97, dan rata-rata akurasi sebesar 0.94. Berdasarkan perbandingan yang dilakukan, metode yang digunakan memiliki performa yang lebih unggul daripada metode FM, DPC, dan CPC.

Pada referensi [15], metode yang digunakan adalah fitur *global-density* yang dipasang pada jaringan CNN (*Convolutional Neural Network*). *Max pooling* pada model dasar diubah dengan *max-average pooling* untuk menjaga ketelitian fitur dan lapisan dekonvolusi ditambahkan untuk mengembalikan detail-detail yang hilang akibat proses *down-sampling* sebelumnya. Eksperimen ini dilakukan dengan 2 dataset, yakni UCF_CC_50 dan ShanghaiTech, dengan menggunakan *framework* Pytorch pada konfigurasi *hardware* GPU GeForce GTX 1080Ti dengan VRAM sebesar 11 GB, CPU ES-2630, RAM 32 GB pada sistem operasi Ubuntu 16.04.3. Digunakan fitur *global-density* yang ditangkap dari sub-tugas klasifikasi kepadatan dan *max-average pooling* untuk menjaga fitur-fitur lebih detail. Pada dataset UCF_CC_50, metode yang digunakan pada penelitian ini menghasilkan MAE sebesar 306.7 dan MSE sebesar 396.3, yang dimana jauh lebih unggul dari metode-metode lainnya yang dirujuk. Sedangkan pada dataset ShanghaiTech *Part A*, didapatkan MAE dan MSE berturut-turut sebesar 86.6 dan 129.7. Pada dataset ShanghaiTech *Part B* mendapatkan MAE dan MSE sebesar 19.3 dan 35.3.

Pada referensi [16], dilakukan analisis sistem pelacak manusia berdasarkan sistem *multi-camera*. Dilakukan analisis 2 metode, yakni Tersentralisasi (*Centralized*) dan Terdistribusi (*Distributed*). Pendekatan Tersentralisasi adalah pendekatan yang mendeteksi dan melacak objek, yang dimana hasil dari deteksi setiap kamera difusi untuk melacak objek (*early fusion*). Sedangkan pendekatan terdistribusi adalah pendekatan yang di mana pendeteksian dan pelacakan objek dilakukan pada masing-masing kamera, kemudian hasilnya difusi untuk membentuk lintasan orang-orang (*late fusion*). Referensi ini membuat sistem hibrid dari yang mengelompokkan kamera menjadi kluster-kluster dengan fusi lokal pusat (*local fusion centers*). Metode ini melacak dengan kamera yang sudah diklusterisasi, pelacakan dilakukan dengan menggunakan pendekatan tersentralisasi (*centralized*) dan menggunakan algoritma *re-identification* untuk membentuk trayek dari orang yang terlacak. Selain sebagai pelacak orang, metode

yang dirujuk ini juga dapat digunakan sebagai sistem pengawasan berbasis AI, kontrol kemacetan, penghitung pedestrian, analisis olahraga.

Pada referensi [17], metode yang digunakan pada penelitian ini adalah YOLOv3 yang dilatih ulang agar hanya mendeteksi objek orang saja, bukan objek lain, sebagai pendeteksi objek, DeepSORT sebagai pelacak objek, dan algoritma pengelompokan untuk mengidentifikasi apakah orang-orang yang berdekatan dalam *frame* termasuk dalam kelompok atau tidak. Algoritma pengelompokan menggunakan *bounding-box* dan *id* yang telah didapatkan dari 2 algoritma sebelumnya, yakni YOLOv3 dan DeepSORT. Penelitian ini memberikan suatu solusi berbasis *Computer Vision* yang mengidentifikasikan jikalau suatu jarak antar orang kurang dari atau sama dengan dari batas yang telah ditentukan, maka orang-orang tersebut dinyatakan sebagai suatu kelompok atau pasangan. Namun, apabila orang-orang tersebut melebihi batas jarak yang ditentukan maka tidak dianggap sebagai suatu kelompok, yakni individu saja. *Training* model *retrained* YOLOv3 menggunakan dataset COCO dan juga dataset *custom* pribadi, hasil *training* pada kedua dataset yang digabungkan menghasilkan nilai *precision*, *recall*, mAP@0.5, dan F1 *score* secara berturut-turut adalah 0.739, 0.809, 0.795, 0.773. Sedangkan hasil uji (*testing*) mendapatkan nilai *precision*, *recall*, mAP@0.5, dan F1-*score* secara berturut-turut adalah 0.746, 0.818, 0.801, dan 0.780. Lalu dilakukan evaluasi dari sistem pengidentifikasian suatu kelompok dengan mengambil video dari CCTV dan menghasilkan nilai *precision* sebesar 0.7083, *recall* sebesar 0.7906, dan F1-*score* sebesar 0.7471.

Pada referensi [18], metode yang digunakan pada penelitian ini adalah YOLOv7 sebagai deteksi objek, BoT-SORT sebagai pelacak objek, dan *Person Re-ID* sebagai *re-identification*. Pembuatan sistem *smart campus* dengan 4 lapisan, yakni lapisan fisik, lapisan jaringan, lapisan integrasi, dan lapisan aplikasi. Pada lapisan fisik, diletakkan kamera IP untuk menangkap video dan mengirimkannya ke server untuk pemrosesan. Kemudian lapisan jaringan yang memungkinkan komunikasi nirkabel antara sub sistem sensor dan *middleware*. Lapisan integrasi yang menyediakan *database* temporal, dan lapisan aplikasi yang mengoleksi data untuk analisis dan visualisasi. Penelitian ini membandingkan performa objek deteksi beberapa seri YOLO pada dataset COCO, yakni YOLOv5n, YOLOR-CSP,

YOLOv7-CSP-X, YOLOv7 tiny, YOLOv7, dan YOLOv7X. Berdasarkan perbandingan yang dilakukan, dipilih model YOLOv7 yang memiliki mAp sebesar 51.4%, GPU APT selama 6.2 ms, dan CPU APT selama 463 ms. Kemudian dilakukan perbandingan beberapa algoritma MOTO seperti SORT, DeepSORT, StrongSORT++, BoT SORT, BoT-SORT-ReID dengan pengujian pada MOT17 dan MOT20. Dipilih algoritma BoT-SORT sebagai pelacak yang memiliki nilai MOTA sebesar 80.6 pada MOT17 dan MOTA sebesar 77.7 pada MOT20. Setelah sistem dibentuk, dilakukan testing secara *real-time* dan mendapatkan kecepatan rata-rata pendeteksian dan pelacakan orang di antara 142.8 ms (7 FPS) dan 263 ms (3.8 FPS).

Pada referensi [19], metode yang digunakan pada penelitian ini ada sistem MCPT yang berbasis *spatio-temporal filtering* dan *group-aware matching* dari fitur-fitur penampilan seseorang. Penelitian ini menggunakan dataset Kumanoto Castle. Sistem yang diciptakan pada penelitian ini terdiri dari 4 proses, yakni proses deteksi objek, *re-identification*, *spatio-temporal filtering*, pembentukan grup. Eksperimen dilakukan dengan menggunakan *framework* Pytorch1.11 yang dijalankan pada GV100 dengan RAM 32 GB. Model deteksi objek yang digunakan pada penelitian ini adalah YOLOX, model MOT yang digunakan pada penelitian adalah ByteTrack, dan model *Re-identification* pada penelitian ini adalah MGN. Penelitian ini meningkatkan akurasi dari antara kamera-kamera menggunakan data yang didapatkan dari spot-spot turis. Dengan menggunakan *spatio-temporal*, sistem dapat berjalan secara efektif pada *scene* yang kompleks. Dan menggunakan penyesuaian *group-aware (group-aware matching)* dapat membuat sistem lebih kokoh dalam melacak kerumunan. Hasil dari sistem yang dibentuk diuji dengan metrik IDF1, IDP, dan IDR yang secara berturut-turut sebesar 78.91, 78.79, dan 79.03, yang di mana ketiga metrik pada metode yang digunakan lebih unggul daripada metode-metode lainnya yang dirujuk pada penelitian ini, yakni HCMIU-CVIP (Deep SORT) dan HCMIU-CVIP (ByteTrack).

Pada referensi [20], metode yang digunakan adalah dengan menggabungkan dinamika kerumunan dengan *Extended CAN (Context Aware Network)*. Metode ini menggunakan model *floor-field* dinamis dan statis serta penambahan suatu ketentuan hukuman (*penalty term*). *Floor-field* statis digunakan sebagai representasi jarak antara hambatan dan tujuan, sedangkan *floor-field* dinamis

digunakan sebagai pengarah arah pergerakan berdasarkan kepadatan manusia di setiap sel yang berdekatan. Dataset yang digunakan pada metode ini adalah CrowdFlow, FDST, Venice, CityStreet, dan UCSD. Seluruh dataset itu mengandung sejumlah gambar orang dan koordinat dari setiap individu sebagai label-label yang sudah dianotasi. Penggunaan *Extended* CAN yang ditambah dengan *floor-field* model dan *penalty term* dapat memiliki akurasi yang lebih baik dalam menghitung banyaknya orang pada kerumunan jikalau dibandingkan dengan model dasarnya, *Extended* CAN konvensional. Pengujian RMSE (*Root Mean Square Error*) *penalty term* pada penelitian ini untuk dataset CrowdFlow, FDST, Venice, CityStreet, dan UCSD berturut-turut adalah 158.3, 7.29, 18, 23.05, 1.02. Metrik yang dihasilkan selalu lebih unggul daripada metode lainnya, seperti MCNN, CAN, ECAN, *Extended* CAN konvensional, dan lain sebagainya.