SISTEM DETEKSI PAKAN UDANG DENGAN METODE DEEP LEARNING

TUGAS AKHIR

Disusun dalam Rangka Memenuhi Salah Satu Persyaratan

untuk Menyelesaikan Program Strata-1 Departemen Teknik Elektro

Fakultas Teknik Universitas Hasanuddin Makassar

DISUSUN OLEH:

JULIAN TULURAN
D411 16 509

DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2020

SISTEM DETEKSI PAKAN UDANG DENGAN METODE DEEP LEARNING

TUGAS AKHIR

Disusun dalam Rangka Memenuhi Salah Satu Persyaratan

untuk Menyelesaikan Program Strata-1 Departemen Teknik Elektro

Fakultas Teknik Universitas Hasanuddin Makassar

DISUSUN OLEH:

JULIAN TULURAN
D411 16 509

DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

LEMBAR PENGESAHAN TUGAS AKHIR

SISTEM DETEKSI PAKAN UDANG DENGAN METODE DEEP LEARNING

Disusun Oleh:

JULIAN TULURAN **D41116509**

Disusun dalam Rangka Memenuhi Salah Satu Pernyataan untuk Menyelesaikan
Program Strata-1 pada Sub-Program Teknik Telekomunikasi
Departemen Elektro Fakultas Teknik Universitas Hasanuddin

Gowa, 25 November 2020

Disahkan Oleh:

Pembimbing I	Pembimbing II
Prof. Dr. Ir. Andani, M.T.	**Dr. Eng Intan Sari Areni, ST.M.T**
NIP. 19601231 198703 1 022	NIP. 19750203 200012 2 002

Mengetahui,

Ketua Departemen Teknik Elektro
Fakultas Teknik Universitas Hasansuddin

DR.Eng. Ir. Dewiani, MT.
NIP. 19691026 199412 2 001

PERNYATAAN KEASLIAN KARYA ILMIAH

Yang bertanda tangan di bawah ini, nama Julian Tuluran, dengan ini menyatakan bahwa skripsi yang berjudul "SISTEM DETEKSI PAKAN UDANG DENGAN METODE DEEP LEARNING", adalah karya ilmiah penulis sendiri, dan belum pernah digunakan untuk mendapatkan gelar apapun dan dimanapun.

Karya ilmiah ini sepenuhnya milik penulis dan semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain yang telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa memiliki kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Gowa, 25 November 2020
Yang membuat pernyataan,

Julian Tuluran
NIM. D411 16 509

KATA PENGANTAR

Salam sejahtera bagi kita semua. Puji dan syukur penulis panjatkan Tuhan Yang Maha Esa yang telah memberikan rahmat-Nya, kekuatan, kesehatan, petunjuk serta kesabaran sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Sistem Deteksi Pakan Udang Dengan Metode *Deep Learning*". Penulis menyadari bahwa masih terdapat banyak kekurangan dalam isi tugas akhir ini sehingga semua kritik dan saran akan sangat bermanfaat untuk penulis agar dapat lebih baik lagi dikemudian hari.

Pembuatan laporan ini berdasarkan perkembangan teknologi yang semakin hari semakin pesat di dunia dan sistem kecerdasan buatan merupakan salah satu bagian dari perkembangan tersebut. Tujuan penulisan laporan tugas akhir ini adalah sebagai salah satu syarat kelulusan pada Pendidikan Strata Satu (S1) di Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin.

Dalam penulisan laporan tugas akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak, untuk itu penulis mengucapkan terima kasih atas bantuan, dukungan, dan doanya. Penulis mengucapkan terima kasih antara lain kepada:

1. Bapak **Agus Tuluran** dan Ibu **Servina Taruk Allo** selaku orang tua penulis yang tidak henti – hentinya memberikan doa dan dukungan dalam bentuk apapun kepada penulis sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak **Prof. Dr. Ir. Andani, M.T.** selaku Dosen Pembimbing I dan Ibu **Dr.Eng. Intan Sari Areni, ST.M.T.** selaku Dosen Pembimbing II penulis yang telah memberikan kritik dan saran bimbingan maupun arahan yang sangat berguna

dalam penyusunan skripsi ini.

3. Bapak **Dr.Eng.. Wardi, S.T, M.Eng** Selaku Dosen Penguji I dan Bapak **Ir. Samuel Panggalo, M.T.** selaku Dosen Penguji II tugas akhir penulis yang telah bersedia meluangkan waktunya untuk menguji penulis dan memberikan saran terkait penyusunan tugas akhir ini.
4. Ibu **Dr. Eng. Ir. Dewiani, MT.** selaku Ketua Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin dan Bapak **Prof. Dr. Baharuddin Hamzah, ST., M.Arch., Ph.D.** selaku Wakil Dekan I Bidang Akademik, Riset dan Inovasi Fakultas Teknik Universitas Hasanuddin.
5. Bapak/Ibu Dosen dan seluruh Staf Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin yang telah banyak memberikan ilmu dan waktu yang tak terbatas selama kuliah dan membantu untuk kelancaran proses penyusunan tugas akhir ini.
6. Kak **Arif Wicaksono** yang senantiasa membantu penulis membuat alat serta memberikan arahan kepada penulis dalam penyelesaian sistem pada tugas akhir ini.
7. **Cantika Glodia** yang selalu menyemangati dan selalu menjadi motivasi bagi penulis untuk menyelesaikan tugas akhir ini.
8. Keluarga **KMKO ELEKTROTEKNIK** dan **KMKO TEKNIK** yang selalu menemani dan memberikan pelayanan di dalam Tuhan.
9. Teman-teman KKN **Bukit Harapan** yang tidak dapat disebutkan satu persatu, yang selalu memberikan kebersamaan dan kebahagiaan kepada penulis selama menjadi bagian dari keluarga ini.
10. Teman-teman **Telekomunikasi** yang telah membantu dan mendukung penulis dalam menghadapi perkuliahan.
11. Teman-teman **NANDEMONAI** yang telah memberikan tawa dan kebersamaan.
12. Teman-teman **EXCITER16** yang tidak dapat disebutkan satu persatu yang selalu memberikan kebersamaan.

Akhir kata penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penulis, dan penulis berharap semoga tugas akhir ini dapat bermanfaat bagi kita semua dan menjadi bahan masukan dalam dunia pendidikan

Gowa, November 2020

Julian Tuluran

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Julian Tuluran, Sistem Deteksi Pakan Udang Dengan Metode Deep Learning (dibimbing oleh Prof. Dr. Ir. Andani, M.T., dan Dr. Eng Intan Sari Areni, ST. MT.).

Udang vaname merupakan salah satu sumber daya perikanan yang ditetapkan sebagai komoditas unggulan pada tahun 2014. untuk memenuhi permintaan udang, maka tambak harus dikelolah dengan intensif yang diikuti peningkatan jumlah benih dan pemberian pakan. Udang termasuk hewan yang tidak efisien dalam memanfaatkan pakan sekitar 70-80% dan sisanya 20-30% terbuang ke lingkungan. Hal ini akan menyebabkan pembusukan sisa pakan dan penurunan kualitas air karena akumulasi bahan organik yang tinggi dan senyawa toksik yaitu nitrit (NO_2) dan amonia (NH_3). Akan tetapi budidaya tambak saat ini masih menggunakan cara manual yakni peninjauan secara langsung untuk pengecekan pakan udang. Untuk itu penulis merancang sistem yang mampu mendeteksi pakan udang dengan metode *Deep Learning* dengan Sistem deteksi Yolo *(You Only Look Once)* versi 3, sehingga dapat mempermudah pengecekan terhadap pakan udang yang berada di dalam air. Yolo bekerja menggunakan algoritma *Convolutional Neural Network* yang menghasilkan *neuron* melalui proses konvolusi, *neuron* tersebut akan dihubungkan kembali pada proses deteksi untuk memprediksi suatu objek. Pertama-tama dilakukan pengumpulan gambar di dalam air, yang digunakan untuk melatih sistem. Gambar kemudian ditandai dengan Yolo *mark.* Kemudian dilakukan *training*, pada tahap ini diekstrak dengan *Convolutional neural network layer* yang hasilnya dijadikan sebagai input ke dalam *Fully Connected Layer* dan luarannya berupa file bobot yang akan digunakan untuk mendeteksi pakan udang. Dari pengujian yang telah dilakukan Sistem menghasilkan nilai mAP yang optimal pada *max batches* 4000-10000 dengan nilai 96-97% dan pengujian sistem terhadap jarak, menghasilkan mAP terbaik pada jarak 25 cm dengan nilai mAP 82.31%. Pada pengujian sistem dengan air tambak udang, air payau, dan air tawar, menghasilkan nilai mAP 0%, 42.8%, dan 64.9%. Yang artinya sistem sulit memprediksi objek pada kondisi air yang keruh. Pada pengujian dengan objek yang menyerupai pakan, terjadi salah prediksi pada pakan ikan, pakan ayam, dan tanah. Sehingga dibutuhkan pengujian lebih lanjut pada ekstraksi fitur di saat melakukan training.

Kata Kunci: Udang Vaname, *Deep Learning*, *Convolutional neural network layer*, Yolo versi 3, *Artificial Intelligence*.

ABSTRACT

Julian Tuluran, Shrimp Feed Detection System Using Deep Learning Method (Supervised by Prof. Dr. Ir. Andani, M.T., and Dr. Eng Intan Sari Areni, ST. MT.).

Vannamei shrimp is one of the fisheries resources designated as a superior commodity in 2014. To meet the demand for shrimp, ponds must be managed intensively, followed by an increase in the number of seeds and feeding. Shrimp including animals that are not efficient in utilizing feed around 70-80% and the remaining 20-30% is wasted into the environment. This will lead to spoilage of the leftover feed and a decrease in water quality due to the high accumulation of organic matter and toxic compounds, namely nitrite (NO2) and ammonia (NH3). However, currently pond cultivation still uses manual methods, namely direct observation to check shrimp feed. For this reason, the authors designed a system that is able to detect shrimp feed using the Deep Learning method with the Yolo (You Only Look Once) detection system version 3, thereby, it can make it easier to check the shrimp feed in the water. Yolo works using a Convolutional Neural Network algorithm that generates neurons through a convolution process, these neurons will be linked back to the detection process to predict an object. First of all, the collection of images in the water is carried out, which is used to train the system. The image is then tagged with the Yolo mark. Then training is carried out, at this stage it is extracted with a Convolutional neural network layer which results are used as input into the Fully Connected Layer and the output is a weight file that will be used to detect shrimp feed. From the tests that have been carried out, the system produces the optimal mAP value at max batches of 4000-10000 with a value of 96-97% and system testing of the distance, produces the best mAP at a distance of 25 cm with a mAP value of 82.31%. In testing the system with shrimp pond water, brackish water, and fresh water, the mAP values were 0%, 42.8%, and 64.9%. Which means that the system is difficult to predict objects in cloudy water conditions. In testing with objects that resemble feed, there were wrong predictions on fish feed, chicken feed, and soil. Therefore, further testing is needed on feature extraction during training.

Keywords: *Vaname Shrimp, Deep Learning, Convolutional neural network layer, Yolo version 3, Artificial Intelligence.*

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Secara geografis Indonesia merupakan sebuah negara kepulauan dengan luas lautannya yang dua per tiga kali lebih besar dibandingkan dengan daratannya. Yang Artinya Indonesia memiliki potensi lahan pesisir untuk menjadi tambak udang terluas di dunia. Hal ini menjadikan masyarakat Indonesia khususnya yang berada di daerah pesisir kebanyakan menjadi seorang nelayan dan juga sebagai petani budidaya tambak. Budidaya udang bagi masyarakat indonesia sangat menguntungkan karena dengan input rupiah dapat menghasilkan *output* dolar. Oleh karena itu maka tidak salah jika departemen kelautan dan perikanan (DKP) telah menetapkan udang sebagai salah satu komoditas unggulan [1].

Untuk memenuhi permintaan udang maka tambak udang harus dikelola secara intensif yang diikuti peningkatan jumlah benih dan pemberian pakan termasuk pakan buatan yang mempengaruhi kondisi lingkungan dan tidak efisien. Udang merupakan hewan yang tidak efisien dalam memanfaatkan pakan, Dalam kondisi tidak terkontrol udang hanya memanfaatkan sekitar 70-80% pakan dan sisanya 20-30% terbuang ke lingkungan sehingga akan berpengaruh terhadap kualitas air pada tambak [2]. Hal ini akan menyebabkan pembusukan sisa pakan dan penurunan kualitas air karena akumulasi bahan organik yang tinggi dan senyawa toksik yang dihasilkan yaitu nitrit (NO_2) dan amonia (NH_3) [3]. Sehingga perlu dilakukan pengontrolan dalam pemberian pakan udang udang. Dimana

frekuensi pemberian pakan dilakukan 4 kali sehari dengan mengecek kondisi pakan pada anco (alat bantu tradisional petambak untuk mengetahui udang memakan pakan yang diberikan) setiap 2-2.5 jam. Jika pakan pada anco habis, maka dosis pemberian pakan udang selanjutnya dapat ditambah secara bertahap sampai dengan 5% dari total pemberian sebelumnya dan jika tidak habis, dosis pemberian pakan selanjutnya dikurangi. [16]

Saat ini, pengontrolan pakan udang masih menggunakan anco. Dimana anco hanya memperkirakan jumlah pakan udang yang berada di dalam tambak. yang artinya anco tidak dapat menentukan pakan udang secara pasti.

Oleh karena itu, pada penelitian ini penulis merancang sistem yang mampu mendeteksi pakan udang di bawah air dengan menggunakan metode *Deep Learning*, untuk mengganti fungsi anco. Sistem deteksi ini akan menganalisa gambar yang diambil dari bawah air, sehingga sistem mampu mengetahui kondisi pakan secara lebih pasti, dan diharapkan sistem deteksi ini dapat mendukung sistem tambak cerdas.

1.2. Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana cara mendeteksi pakan udang di dalam air menggunakan metode *Deep Learning*?
2. Bagaimana kinerja dari sistem deteksi pakan udang di dalam air menggunakan metode *Deep Learning*?

1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Merancang sistem yang mampu mendeteksi pakan udang di dalam air menggunakan metode *Deep Learning*.
2. Mengetahui hasil pengujian dari sistem deteksi pakan udang di dalam air dengan metode *Deep Learning*.

1.4. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Pengambilan data dilakukan dengan format gambar.
2. Objek penelitian berupa pakan udang berbentuk pellet.
3. Penelitian dilakukan di dalam pada wadah berlatar biru berukuran $1m \times 2m \times 50\ cm$, tanpa menggunakan aerator.
4. Perancangan sistem deteksi dibuat dengan menggunakan metode *Deep Learning* algoritma YOLO v3 untuk pengenalan objek berupa pakan udang.

1.5. Manfaat Penelitian

Dengan adanya penelitan ini maka manfaat yang diharapkan, antara lain:

1. Bagi peneliti, penelitian ini diharapkan dapat menjadi referensi tambahan dalam pengembangan sistem tambak cerdas, serta dapat digunakan untuk menambah pengetahuan di bidang *Deep Learning* dalam mendeteksi objek-objek yang berada di bawah air.
2. Bagi masyarakat khususnya petani tambak, penelitian ini diharapkan dapat digunakan untuk membantu petani dalam mengawasi pakan udang.

3. Bagi penulis, penelitian ini diharapkan dapat menjadi tolak ukur kemampuan penulis, serta menjadi bahan evaluasi mengenai penerapan teori yang ada di perkuliahan.

1.6. Sistematika Penulisan

Agar pembahasan yang disajikan lebih sistematis, maka Tugas Akhir ini dibagi ke dalam lima bab. Isi masing-masing dari bab diuraikan secara singkat;

BAB 1 PENDAHULUAN

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan dan manfaat penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

BAB 2 TINJAUAN PUSTAKA

Bab ini berisi tentang teori dasar yang berhubungan dengan penulisan laporan dan menunjang penelitian sistem deteksi pakan udang di dalam air dengan metode *Deep Learning.*

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi proses perancangan sistem, dan menjelaskan metode yang digunakan untuk melihat kinerja sistem untuk mendeteksi pakan udang di dalam air dengan metode *Deep Learning*

BAB 4 HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil, dan pembahasan masalah dari penelitian sistem deteksi pakan udang di dalam air dengan metode *Deep Learning.*

BAB 5 PENUTUP

Bab ini merupakan penutup yang berisi kesimpulan tentang hasil pemecahan masalah yang diperoleh selama penyusunan tugas akhir, serta tambahan beberapa saran untuk pengembangan penelitian lebih lanjut terkhusus di bidang *Deep Learning* pada waktu yang akan datang.

BAB 2

TINJAUAN PUSTAKA

2.1. Udang Vaname

Udang vaname *(Litopenaeus vannamei)* berasal dari daerah subtropis pantai barat Amerika, mulai dari Teluk California di Mexico bagian utara sampai ke pantai barat Guatemala, El Salvador, Nicaragua, Kosta Rika di Amerika Tengah hingga ke Peru di Amerika Selatan.

Pertumbuhan udang vaname dipengaruhi dua faktor yaitu frekuensi molting/ganti kulit (waktu antara molting) dan pertumbuhan pada setiap molting. Tubuh udang mempunyai karapas/kulit luar yang keras, sehingga pada setiap kali berganti kulit, karapas terlepas dan akan membentuk karapas baru. Ketika karapas masih lunak, udang berpeluang untuk dimangsa oleh udang lainnya.

Udang merupakan organisme pemakan segala *(omnivora)*. Pada habitatnya, udang vaname memakan jasad *renik/krustasea* kecil, *amphipoda* dan *polychaeta*. Udang vaname tidak makan sepanjang hari, tetapi hanya beberapa waktu saja dalam sehari. Nafsu makan tergantung oleh kondisi lingkungan dan laju konsumsi pakan akan meningkat pada kondisi lingkungan optimum [4].

2.2. Klasifikasi Udang Vaname

Klasifikasi udang vaname adalah sebagai berikut [5]:

Kingdom : Animalia

Filum : Arthropoda

Kelas : Malacostraca

Ordo : Decapoda

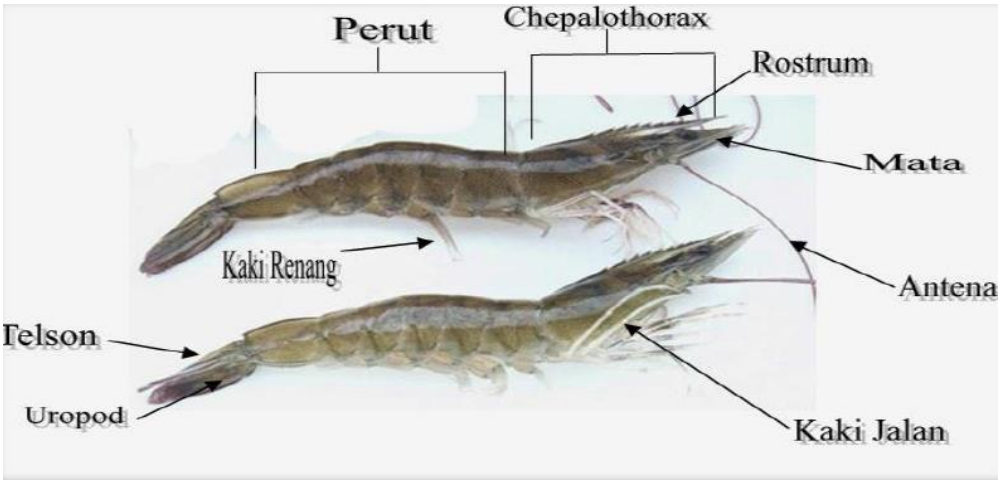
Famili : Penaeidae

Genus : *Litopenaeus*

Spesies : *Litopenaeus vaname*

2.3. Morfologi Udang Vaname

Udang vaname termasuk dalam famili *Penaeidae,* karena itu sifat umum morfologi sama dengan udang windu. Morfologi udang vaname diperlihatkan pada **Gambar 2.1.** Tubuh udang putih vaname secara morfologis dapat dibedakan menjadi dua bagian yaitu *cephalothorax* atau bagian kepala dan dada serta bagian abdomen atau perut. Bagian *chepalothorax* terlindung oleh chitin yang tebal yang dinamakan carapace. Kulit chitin pada udang penaeid, akan selalu mengalami pergantian kulit setiap kali tubuhnya akan membesar, setelah itu kulitnya akan mengeras kembali. [5].



Gambar 2.1. Morfologi Udang Vaname [5].

2.4. Biologi Udang Vaname

Udang vaname (Litopenaeus vaname) merupakan salah satu jenis udang yang memiliki pertumbuhan cepat dan nafsu makan yang tinggi, namun ukuran yang dicapai pada saat dewasa lebih kecil dibandingkan udang windu (Penaeus monodon). Habitat aslinya adalah di perairan samudera pasifik, tetapi spesies ini dapat dibudidayakan dengan baik di Indonesia [4]. Informasi ilmiah lebih rinci mengenai udang ini dijabarkan dalam biologi udang putih vaname, meliputi: taksonomi dan anatomi, morfologi, habitat dan daur hidup, pakan dan kebiasaan makan.

2.5. Pengolahan Citra

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari *webcam*). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer. Secara matematis, citra merupakan fungsi kontinu (*continue*) dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Reprsentasi dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi citra. Sebuah citra digital dapat diwakili oleh sebuah matriks dua dimensi $f(x,y)$ yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel (*pixel = picture element*) atau elemen terkecil dari sebuah citra [7].

Pada aplikasi pengolahan citra digital pada umumnya, citra digital dapat dibagi menjadi 3, *color image, black and white image* dan *binary image* [7]**:**

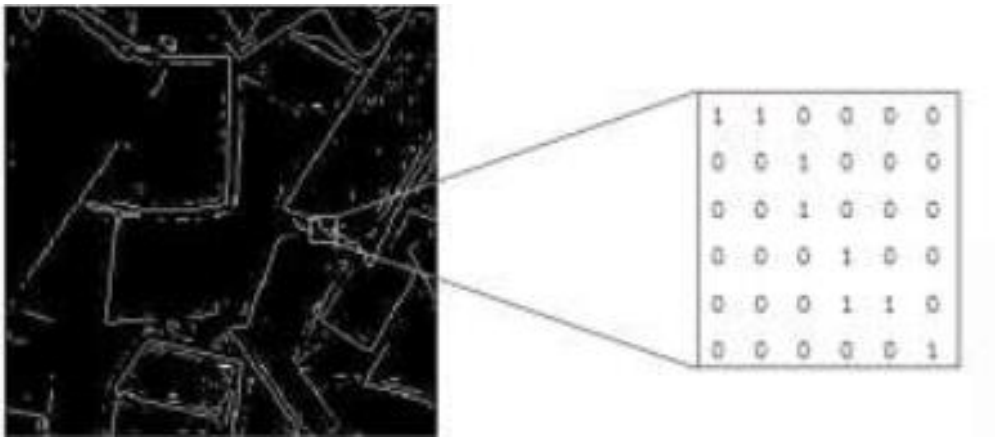
1. *Color Image* atau RGB (*Red, Green, Blue*). Pada *color image* ini masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*Red*), hijau (*Green*) dan biru (*Blue*). Jika masing-masing warna memiliki range 0 - 255, maka totalnya adalah 2553 = 16.581.375 variasi warna berbeda pada gambar, dimana variasi warna ini cukup untuk gambar apapun. Karena jumlah bit yang diperlukan untuk setiap pixel, gambar tersebut juga disebut gambar-bit warna. *Color image* ini terdiri dari tiga matriks yang mewakili nilai-nilai merah, hijau dan biru untuk setiap pikselnya, seperti yang ditunjukkan **Gambar 2.2**

Gambar 2.2. *Color Image* [7]

2. *Black and White*. Citra digital *black and white* (*grayscale*) setiap pikselnya mempunyai warna gradasi mulai dari putih sampai hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit, atau 1 *byte*. Rentang warna pada *black and white* sangat cocok digunakan untuk pengolahan file gambar. Salah satu bentuk fungsinya digunakan dalam kedokteran (*X-ray*). Contoh *grayscale image* dapat dilihat pada **Gambar 2.3.**

Gambar 2.3. *Grayscale Image* [7]

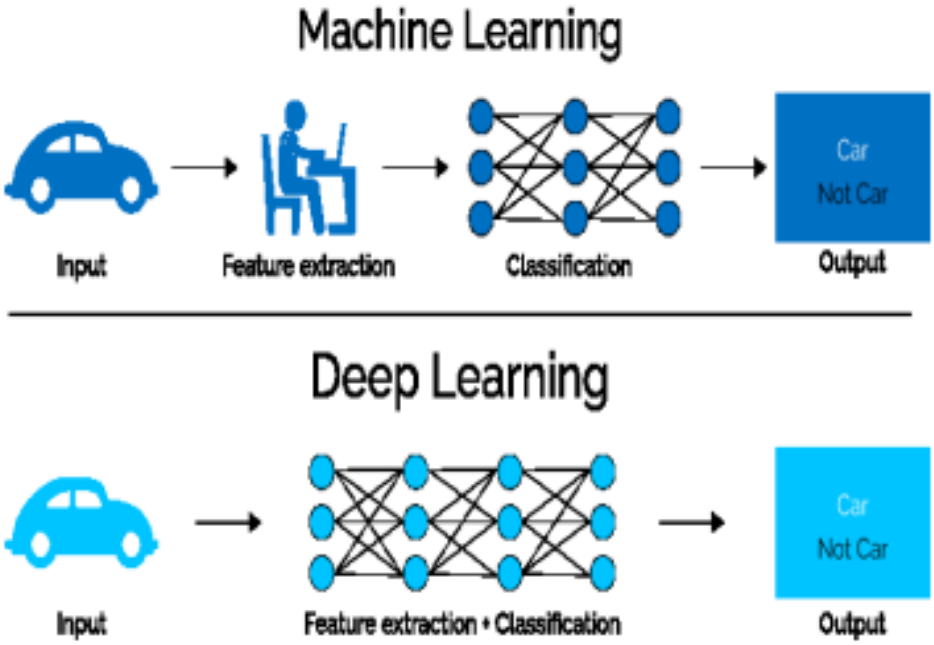
3. *Binary Image*. Setiap piksel hanya terdiri dari warna hitam atau putih, karena hanya ada dua warna untuk setiap piksel, maka hanya perlu 1 bit per piksel (0 dan 1) atau apabila dalam 8 bit (0 dan 255), sehingga sangat efisien dalam hal penyimpanan. Contoh *binary image* dapat dilihat pada **Gambar 2.4.** Gambar yang direpresentasikan dengan biner sangat cocok untuk teks (dicetak atau tulisan tangan), sidik jari (*finger print*), atau gambar arsitektur. *Binary image* merupakan hasil pengolahan dari *black and white image*.



Gambar 2.4. *Binary Image* [7].

2.6. *Deep Learning*

Deep learning merupakan area baru dalam bidang ilmu *machine learning*. *Deep learning* pertama sekali dikemukakan oleh Bapak *neural network* yaitu Prof. Dr. Geoffrey Hinton dari Toronto University, yang dipublikasikan di dalam karya tulisnya yang berjudul *A fast learning algorithm for deep belief nets*. Ketenaran *Deep Learning* baru bermula pada tahun 2012 ketika *deep learning* berhasil menyelesaikan klasifikasi lebih dari 1,2 juta gambar dari kompetisi *ImageNet*. Keberhasilan ini menjadikan industri IT skala besar seperti Google, Facebook, Amazon, dll kembali tertarik dengan bidang Kecerdasan Buatan. Google Brain ditahun yang sama juga berhasil melaksanakan *"Cat Experiment"* menggunakan dataset yang sangat besar dan dalam jenis *unsupervised learning*. Google *Brain* berhasil mengidentifikasi wajah kucing dengan pembelajaran yang mendalam *Deep Learning* merupakan teknik dalam *machine learning* yang memiliki arsitektur yang lebih "*deep*" dibanding dengan teknik *machine learning* lainnya dalam menyelesaikan masalah prediksi, maupun klasifikasi. Perbedaan *machine learning* dengan *deep learning* dapat dilihat secara visual pada **Gambar 2.5** [8].



Gambar 2.5. Perbedaan *Machine Learning* dan *Deep Learning* [8].

Deep Learning merupakan cabang ilmu dari *Machine Learning* yang berbasis Jaringan Syaraf Tiruan (JST) atau dapat dikatakan perkembangan dari JST yang mengajarkan komputer untuk dapat melakukan tindakan yang dianggap alami oleh manusia. Misalnya yaitu belajar dari contoh. Dalam *Deep Learning*, sebuah komputer dapat belajar mengklasifikasi secara langsung dari gambar, suara, teks, atau video sekalipun. Sebuah komputer seperti dilatih dengan menggunakan data set berlabel dan jumlahnya sangat besar yang kemudian dapat mengubah nilai piksel dari sebuah gambar menjadi suatu representasi internal atau *feature vector* yang dimana pengklasifikasiannya dapat digunakan untuk mendeteksi atau mengklasifikasi pola pada masukan input.

Metode *deep learning* adalah metode pembelajaran dengan beberapa tingkat representasi, dimana representasi dapat membentuk arsitektur jaringan syaraf yang mempunyai banyak *layer* (lapisan). Lapisan pada deep learning terbagi menjadi tiga

bagian yaitu, *input layer, hidden layer, dan output layer*. Pada hidden layer dapat dibuat banyak lapis atau berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar dapat meminimalisir *error* pada *output*. Semakin banyak *layer* yang dihasilkan, maka akan semakin kecil *error* yang dihasilkan, sehingga dapat menghasilkan akurasi yang lebih bagus atau lebih tinggi.

Bobot adalah koneksi antar lapisan yang berupa nilai yang menentukan fungsi *input-output* dari mesin. Bobot adalah parameter penyesuaian yang diatur oleh mesin untuk mengukur kesalahan antara nilai *output* dan pola nilai yang diinginkan pada pembelajaran. Sehingga, nilai bobot inilah yang diatur mesin untuk mengurangi kesalahan yang mungkin terjadi. Dalam sistem *deep learning*, kemungkinan ada ratusan juta bobot yang dapat diatur. Untuk menyesuaikan vektor bobot dengan benar, algoritma menghitung nilai gradien vektor untuk setiap bobot berdasarkan jumlah kesalahan yang meningkat atau menurun jika bobot meningkat dalam jumlah kecil [9].

Deep Learning adalah teknik dalam *neural network* yang menggunakan teknik tertentu seperti *Restricted Boltzmann Machine* (RBM) untuk mempercepat proses pembelajaran dalam *neural network* yang menggunakan lapis yang banyak atau lebih dari 7 lapis. Dengan adanya *deep learning*, waktu yang dibutuhkan untuk training akan semakin sedikit karena masalah hilangnya gradien pada propagasi balik akan semakin rendah. Beberapa jenis *deep learning* antara lain *Deep Autoencoder, Deep Belief Nets, Convolutional Neural Network*, dan lain lain. Tabel Perbandingan beberapa algoritma dengan Metode *Deep Learning* dilihat pada **Tabel 2.1** [10].

Tabel 2.1 Perbandingan beberapa Metode *Deep Learning* [9].

Metode	Kelebihan	Kekurangan
K-Nearest Neighbour	KKN mempunyai beberapa kelebihan bahwa metode ini tangguh untuk melakukan *training* data yang *noisy* dan efektif apabila data latihnya besar.	• KKN perlu menentukan nilai dari parameter K (jumlah dari tetangga terdekat). Pembelajaran berdasarkan jarak. Tidak jelas mengenai jenis jarak. apa yang harus digunakan dan • atribut mana yang harus digunakan untuk mendapatkan hasil yang terbaik.
YOLO (You Only Look Once) versi 3	• Yolo merupakan algoritma yang cukup mudah dioperasikan, karena parameter pada algoritma Yolo dapat langsung diubah melalui file konfigurasi • Yolo bekerja dengan lebih cepat dengan tingkat akurasi yang lebih baik. • (*Real-time)*	• Objek dilabeli satu persatu sehingga jika pada gambar terdapat banyak objek akan sulit melabeli objek • Yolo kurang cocok digunakan untuk mendeteksi objek kecil.

Lanjutan Tabel 2.1 Perbandingan beberapa Metode Deep Learning [9].

Metode	Kelebihan	Kekurangan
Klasifikasi Naive Bayes	Mampu bekerja tangguh terhadap da ta-data yang terisolasi, biasanya data *outliner*. Dapat menangani nilai atribut yang salah dengan mengabaikan data latih selama proses pembangunan model dan prediksi • Atribut mempunyai korelasi bisa mendegradasi kinerja klasifikasi Naive Bayes karen asumsi independensi atribut sudah tidak ada.	• Tidak berlaku jika probabilitas kondisionalnya adalah nol, apabila nol maka probabilitas prediksi akan nol juga. • Diasumsikan Variabel bebas.
Convolutional Neural Network (CNN)	• *Layer* dapat ditentukan oleh peneliti. • Banyaknya *hidden layer* membuat hasil klasifikasi menjadi lebih baik tingkat akurasinya. • Dilakukan pelabelan terlebih dulu untuk data latih sehingga	• CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara

Lanjutan Tabel 2.1 Perbandingan beberapa Metode Deep Learning [9].

Metode	Kelebihan	Kekurangan
	pada data uji hasilnya akan lebih akurat	
Support Vector Machine (SVM)	• Generalisasi: kemampuan suatu metode untuk mengklasifikasikan suatu *pattern*, yang tidak termasuk data yang dipakai dalam fata latihnya. *Curse of dimensionally*: Masalah yang dihadapi suatu metode *pattern recognition* dalam mengestimasikan • parameter (jumlah *hidden neuron* pada *neural network*, *stopping criteria* dalam proses pembelajaran, karena data sampel yang relatif sedikit. • dapat dirumuskan dalam QP *problem*	• Sulit dipakai dalam skala besar (jumlah sampel) SVM dikembangkan untuk klasifikasi dua kelas.

2.7. *Neural Network*

Jaringan Syaraf Tiruan atau *Artificial Neural Network* adalah teknik dalam *Machine Learning* yang menirukan syaraf manusia yang merupakan bagian fundamental dari otak. *neural network* terdiri atas lapis masukan (*input layer*) dan lapis keluaran (*output layer*). Setiap lapis terdiri atas satu atau beberapa unit *neuron* yang mempunyai sebuah fungsi aktivasi yang menentukan keluaran dari unit tersebut. Dapat dilakukan penambahan lapis tersembunyi (*hidden layer*) untuk menambah kemampuan dari *neural network* tersebut. *neural network* bisa dilatih dengan menggunakan data training. Semakin banyak data training maka akan semakin bagus unjuk kerja dari *neural network* tersebut. Namun, kemampuan *neural network* juga terbatas pada jumlah lapisan, semakin banyak jumlah lapisan semakin tinggi kapasitas *neural network* tersebut. Semakin banyak lapisan juga membawa kekurangan yaitu semakin banyaknya jumlah iterasi atau training yang dibutuhkan. Untuk mengatasi hal ini, dikembangkanlah teknik *Deep Learning*. Beberapa aplikasi *neural network* antara lain untuk *Principal Component Analysis*, 2 regresi, klasifikasi citra, dan lain-lain [10].

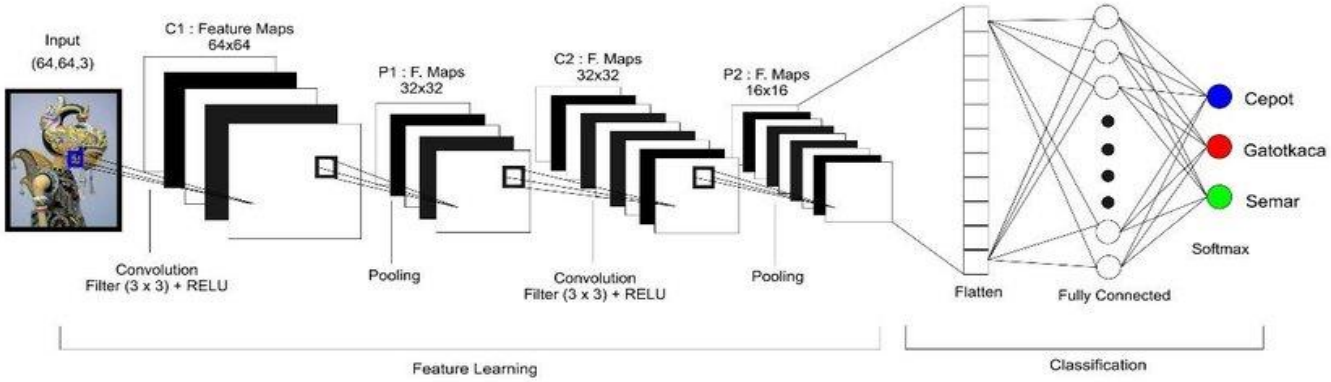
2.8. *Convolutional Neural Network*

Convolutional Neural Network merupakan salah satu algoritma dari deep learning yang merupakan hasil pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk melakukan olah data menjadi bentuk dua dimensi, misalnya yaitu: gambar atau suara. *Convolution neural network* digunakan untuk melakukan klasifikasi data yang berlabel dengan menggunakan metode *supervised learning* yang cara kerjanya dari *supervised learning* adalah terdapat data yang dilatih dan

terdapat variabel yang ditargetkan sehingga tujuan dari metode ini yaitu mengelompokkan suatu data ke data yang sudah ada.

Convolution neural network sering digunakan untuk mengenali benda berdasarkan penampakan dan melakukan deteksi dan melakukan segmentasi objek. *Convolution neural network* belajar langsung melalui data citra, sehingga dapat menghilangkan ekstraksi ciri dengan cara manual. Penelitian awal yang menjadi dasar penemuan ini yaitu pertama kali dilakukan oleh Hubel dan Wiesel yang melakukan penelitian *visual cortex* pada indera penglihatan kucing. *Visual cortex* pada hewan sangat *powerful* kemampuannya dalam sistem pemrosesan visual yang pernah ada. Sehingga, banyak penelitian yang terinspirasi oleh cara kerjanya dan menghasilkan banyak model-model baru yang beberapa diantaranya yaitu, *Neocognitiron*, HMAX, LeNet-5, dan AlexNet.

CNN juga merupakan saraf yang dikhususkan untuk memproses data yang memiliki struktur kotak (*grid*). Sebagai contoh yaitu berupa citra dua dimensi. Nama konvolusi merupakan operasi dari aljabar linear yang mengalikan matriks dari filter pada citra yang akan diproses. Proses ini disebut dengan lapisan konvolusi dan merupakan salah satu jenis dari banyak lapisan yang bisa dimiliki dalam suatu jaringan. Meskipun begitu, lapisan konvolusi ini merupakan lapisan utama yang paling penting digunakan. Jenis lapisan yang lain yang biasa digunakan adalah *Pooling Layer*, yakni lapisan yang digunakan untuk mengambil suatu nilai maksimum atau nilai rata-rata dari bagian-bagian lapisan piksel pada citra. gambaran umum arsitektur *Convolution Neural Network* diperlihatkan pada **Gambar 2.6.**



Gambar 2.6. Contoh Jaringan *Convolution Neural Network* [11].

Pada **Gambar 2.6** setiap lapisan *input* yang dimasukkan mempunyai susunan neuron 3 dimensi, yaitu lebar, tinggi, dan kedalaman) Lebar dan tinggi yaitu ukuran lapisan, sedangkan untuk kedalaman yaitu mengacu pada jumlah lapisan. Setiap besaran yang didapat tergantung dari hasil filtrasi dari lapisan sebelumnya dan banyaknya filter yang digunakan. Model jaringan seperti ini sudah terbukti efektif dalam menangani permasalahan klasifikasi citra. Sebuah *convolution neural network* mampu memiliki puluhan hingga ratusan lapisan yang masing-masing lapusan mempelajari deteksi berbagai gambar. Pengolahan citra diterapkan pada setiap citra latih pada resolusi yang berbeda, dan *output* dari masing-masing data gambar yang diolah dan digunakan sebagai *input* ke lapisan berikutnya. Pengolahan citra dapat dimulai sebagai fitur yang sederhana, seperti ukuran kecerahan dan tepi atau meningkatkan kekompleksan pada fitur secara unik untuk menentukan objek sesuai ketebalan lapisan [9].

Secara umum tipe lapisan *convolution neural network* dibagi menjadi dua bagian, yaitu [9]:

a. *Layer* Ekstraksi Fitur (*feature extraction layer*)

Gambar yang letaknya ada di awal arsitektur yang tersusun atas beberapa lapisan dan di setiap susunan lapisannya atas *neuron* yang terkoneksi pada daerah lokal (*local region*) dari lapisan sebelumnya. Lapisan pada jenis pertama yaitu adalah *convolutional layer* dan lapisan kedua adalah *pooling layer*. Pada setiap lapisan diberlakukan fungsi aktivasi dengan posisinya yang berselang-seling antara jenis pertama dan jenis kedua. Lapisan ini menerima input gambar secara langsung dan memprosesnya sampai menghasilkan *output* berupa vektor untuk diolah di lapisan berikutnya.

b. *Layer* Klasifikasi (*classification layer)*

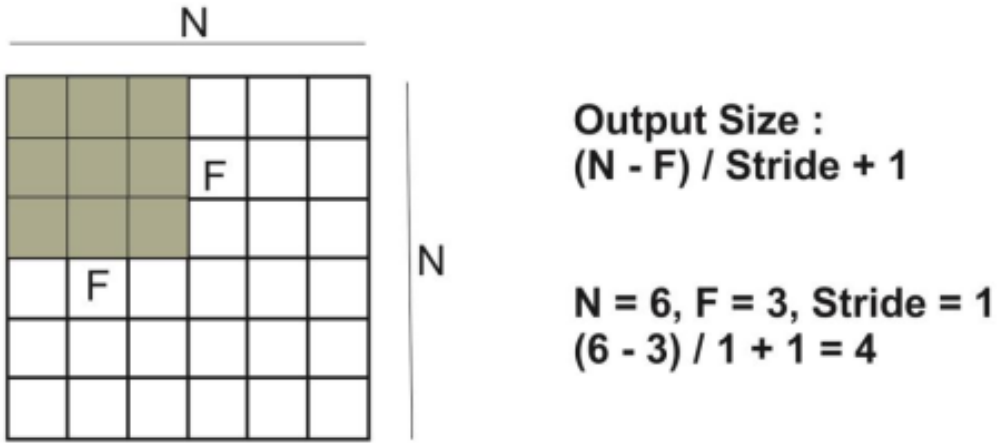
Layer ini tersusun atas beberapa lapisan yang di setiap lapisan tersusun atas neuron yang terkoneksi secara penuh (*fully connected*) dengan lapisan yang lainnya. Layer ini menerima input dari hasil output layer ekstraksi fitur gambar berupa vektor yang kemudian ditransformasikan seperti pada *Multi Neural Network* dengan tambahan beberapa *hidden layer*. Hasil *output* berupa akurasi kelas untuk klasifikasi.

2.9. *Convolutional Layer*

Convolutional Layer bagian yang melakukan operasi konvolusi yaitu mengkombinasikan linier *filter* terhadap daerah lokal. *Layer* ini yang pertama kali

menerima gambar yang diinputkan pada arsitektur. Bentuk *layer* ini adalah sebuah *filter* dengan panjang (pixel), tinggi (pixel), dan tebal sesuai dengan *channel image* data yang diinputkan. Ketiga *filter* ini akan bergeser ke seluruh bagian gambar. Pergeseran tersebut akan melakukan operasi "dot" antara input dan nilai dari *filter* tersebut sehingga akan menghasilkan output yang disebut sebagai *activation map* atau *feature map*. **Gambar 2.7** menampilkan proses konvolusi yang ada di dalam *convolutional layer* dan **Gambar 2.8** adalah cara menghitung nilai konvolusinya [12].

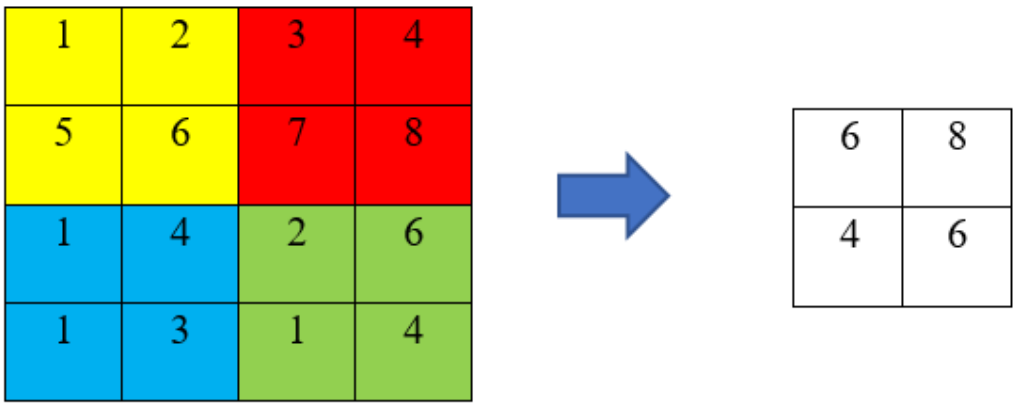
Gambar 2.7. Proses Konvolusi [12]



Gambar 2.8. Rumus menghitung Konvolusi [12]

2.10. *Pooling Layer*

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling Layer* biasanya berada setelah *conv*. Pada dasarnya *pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan secara bergantian bergeser pada seluruh *area feature map*. Dalam *pooling layer* terdapat dua macam pooling yang biasa digunakan yaitu *average pooling* dan *max-pooling.* Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan Pooling yang dimasukkan di antara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume *output* pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *Overfitting*. Lapisan pooling bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan pooling umumnya dengan menggunakan *filter* dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Contoh operasi *max-pooling* dapat dilihat pada **Gambar 2.9**



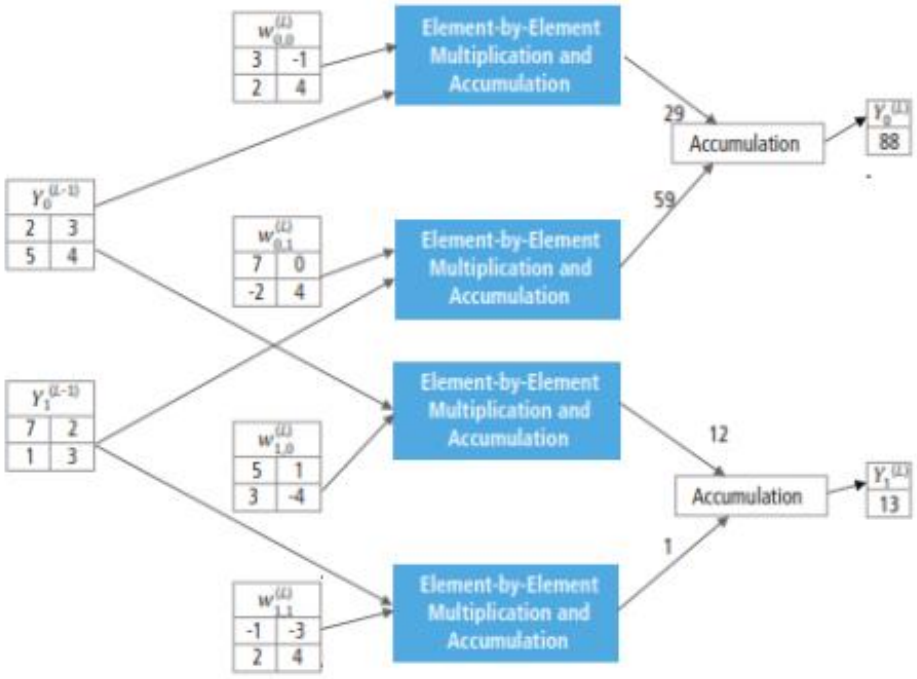
Gambar 2.9. Operasi *Max Pooling*

Berdasarkan **Gambar 2.9** menunjukan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan *pooling* akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Jika dilihat dari **Gambar 2.9** *operasi max-pooling* dengan menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2 [11].

2.11. *Fully-Connected Layer*

Fully-Connected Layer adalah sebuah lapisan dimana sesuai *neuron* aktivasi dari lapisan sebelumnya terhubung semua dengan neuron di lapisan selanjutnya sama seperti halnya dengan *neural network* biasa. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data, agar data dapat diklasifikasikan secara linear.

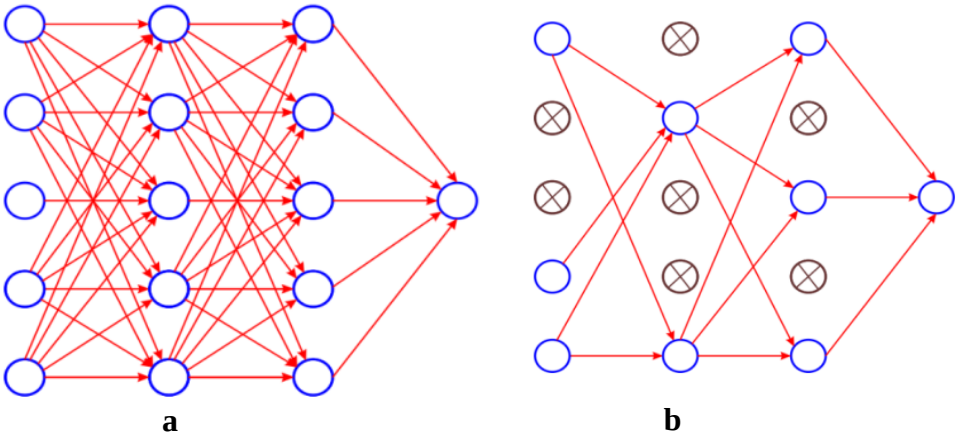
Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah *neuron* di lapisan konvolusi terhubung hanya ke daerah tertentu pada input, sementara lapisan *Fully-Connected* memiliki *neuron* yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak. begitu berbeda. Pada **Gambar 2.10** diperlihatkan proses *fully-connected layer* [11].



Gambar 2.10. *Processing of a Fully-Connected Layer* [11]

2.12. *Dropout*

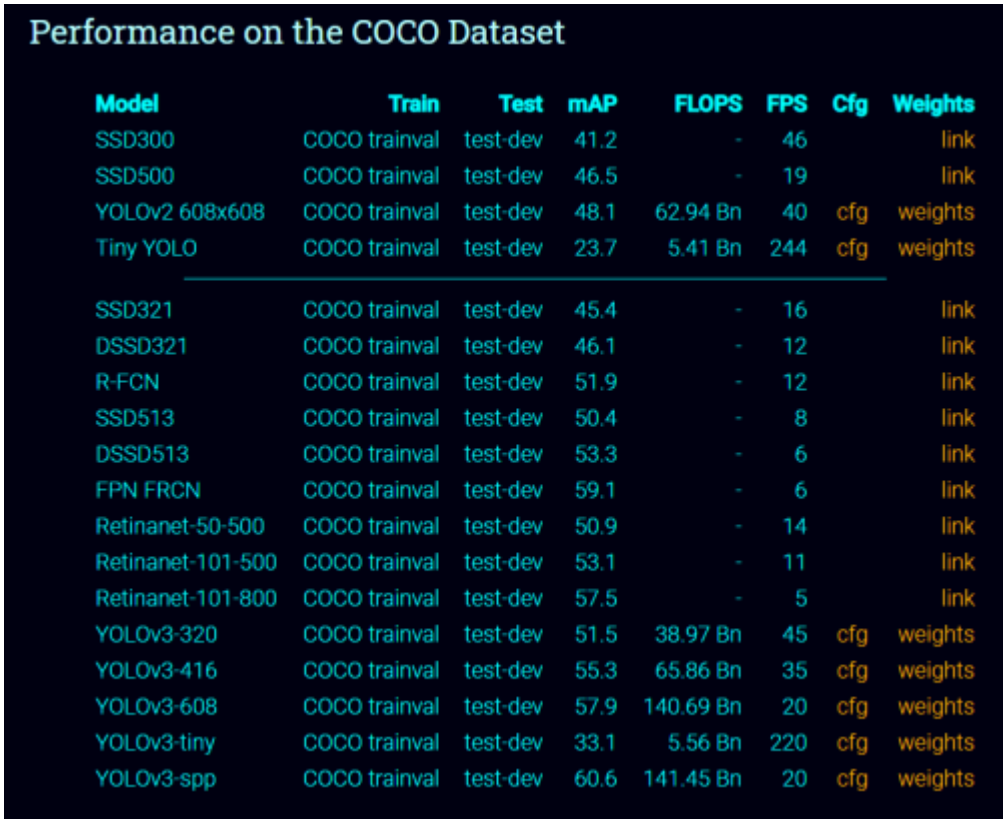
Dropout merupakan salah satu usaha untuk mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Overfitting* adalah kondisi dimana hampir semua data yang telah melalui proses *training* mencapai persentase yang baik, tetapi terjadi ketidaksesuaian pada proses prediksi. Dalam sistem kerjanya, *Dropout* menghilangkan sementara suatu *neuron* yang berupa *Hidden Layer* maupun *Visible Layer* yang berada di dalam jaringan. Contoh *Dropout* diperlihatkan pada **Gambar 2.11** [13].



Gambar 2.11. (a) Sebelum *Dropout.* (b) Sesudah *Dropout* [13]

2.13. YOLO V3 (*You Only Look Once)*

Yolo merupakan algoritma pendeteksi objek, yang bekerja secara *real-time* dan cepat. Yolo menggunakan pendekatan yang berbeda untuk mendeteksi objek. Dengan menerapkan jaringan neural tunggal ke gambar penuh. Jaringan ini membagi gambar menjadi beberapa wilayah dan memprediksi *Bounding Box* dan probabilitas untuk setiap wilayah. *Bounding Box* ini diberi bobot oleh probabilitas yang diprediksi. YOLO meiliki 3 versi yaitu YOLO V1 merupakan versi awal YOLO yang sudah jarang digunakan, YOLO V2 yang merupakan versi kedua dari YOLO yang masih banyak digunakan oleh praktisi untuk melakukan pendeteksian secara real-time, dan YOLO V3 yang merupakan versi paling update dan memiliki tingkat akurasi yang tinggi [14].



Performance on the COCO Dataset

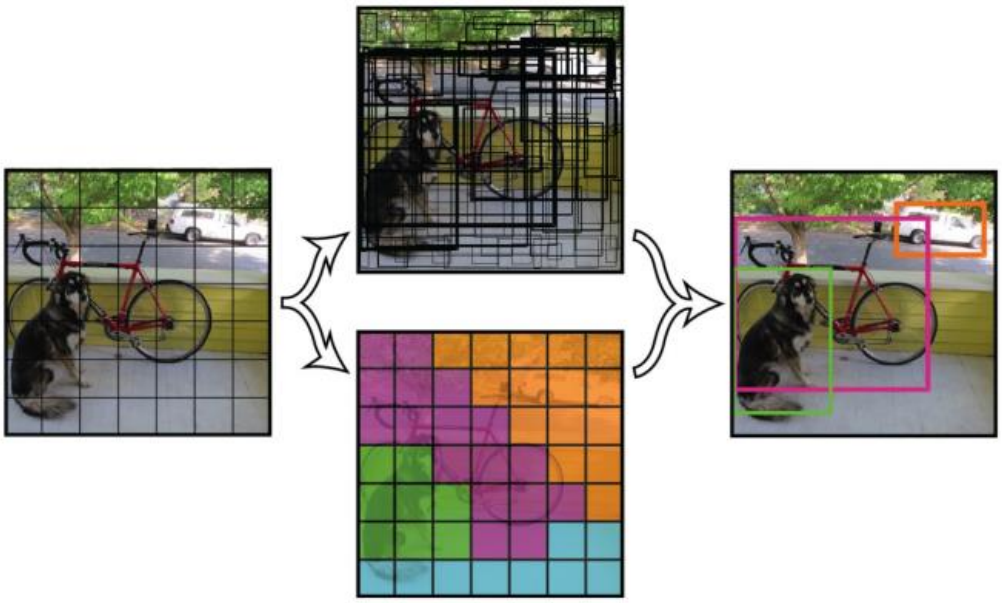
Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Gambar 2.12. Perbandingan kinerja deteksi pada *COCO Dataset* [15]

Pada **Gambar 2.12** dapat terlihat bahwa mAP (*mean Average precision*) pada YOLO lebih tinggi dibandingkan algoritma deteksi lain, artinya akurasi yang akan di dapat kan jika menggunakan YOLO akan lebih baik jika dibandingkan dengan algoritma lain [15].

Kebanyakan sistem deteksi sebelumnya menggunakan pengklasifikasian atau *localizer* untuk melakukan deteksi dengan menerapkan model ke gambar di beberapa lokasi dan memberi nilai *confidence* pada gambar sebagai bahan untuk pendeteksian. YOLO menggunakan pendekatan yang sangat berbeda dengan algoritma sebelumnya, yakni menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas, untuk setiap kotak wilayah

pembatas ditimbang probabilitasnya untuk mengklasifikasian sebagai objek atau bukan [6]. Ilustrasi dari proses deteksi YOLO dapat dilihat pada **Gambar 2.13.**



Gambar 2.13. Ilustrasi proses deteksi YOLO [6].

YOLO memiliki arsitektur yang sederhana, yaitu jaringan saraf tiruan. Jaringan saraf ini hanya menggunakan jenis lapisan standar: konvolusi dengan kernel 3 × 3 dan *max-pooling* dengan 2 × 2 kernel. Lapisan konvolusional terakhir memiliki 1 × 1 kernel digunakan untuk mengecilkan data, misalnya ke bentuk 13 × 13 × 125. 13×13 ini merupakan ukuran *grid* yang terbagi dibagi gambar. 125 (5 sel grid × 25 element data) merupakan jumlah Channel untuk setiap *grid*. dan berisi data untuk kotak pembatas dan prediksi kelas [6]. Ilustrasi lapisan konvolusi YOLO V3 dapat dilihat pada **Gambar 2.13.**

Layer	kernel	stride	output shape
Input			(416, 416, 3)
Convolution	3×3	1	(416, 416, 16)
MaxPooling	2×2	2	(208, 208, 16)
Convolution	3×3	1	(208, 208, 32)
MaxPooling	2×2	2	(104, 104, 32)
Convolution	3×3	1	(104, 104, 64)
MaxPooling	2×2	2	(52, 52, 64)
Convolution	3×3	1	(52, 52, 128)
MaxPooling	2×2	2	(26, 26, 128)
Convolution	3×3	1	(26, 26, 256)
MaxPooling	2×2	2	(13, 13, 256)
Convolution	3×3	1	(13, 13, 512)
MaxPooling	2×2	1	(13, 13, 512)
Convolution	3×3	1	(13, 13, 1024)
Convolution	3×3	1	(13, 13, 1024)
Convolution	1×1	1	(13, 13, 125)

Gambar 2.13. Ilustrasi lapisan konvolusi YOLO V3 [6]