

BAB II

LANDASAN TEORI

2.1 *Machine Learning*

Machine Learning (ML) atau pembelajaran mesin merupakan bagian dari kecerdasan buatan. *Machine learning* ini adalah studi ilmiah tentang algoritma dan model statistik yang digunakan sistem komputer untuk melakukan tugas tertentu tanpa menggunakan instruksi eksplisit, dengan mengandalkan pola dan inferensi sebagai gantinya (Pratama, 2020). Menurut Arthur (dalam Utama, 2018) *machine learning* merupakan metode berbasis komputer dimana komputer diberikan kemampuan untuk belajar dengan bantuan data tanpa harus terprogram terlebih dahulu. Sedangkan menurut Müller dan Guido (2016 dalam Pambudi et al., 2020) *machine learning* yaitu algoritma ekstraksi data yang merupakan irisan dari ilmu statistik, kecerdasan buatan (artificial intelligence) dan ilmu komputer (computer science).

Algoritma *machine learning* membangun model matematika berdasarkan data sampel, yang dikenal sebagai “*data training*” untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukan tugasnya (Bishop, 2006 dalam Pratama, 2020) dengan kata lain *machine learning* fokus pada program komputer yang secara otomatis dapat meningkatkan kinerjanya melalui/berdasarkan pengalaman.

Secara umum, *machine learning* dibagi menjadi 3, yaitu:

a. *Supervised Learning*

Model *supervised learning* yaitu teknik dimana model diajarkan dengan pengetahuan sehingga dapat memprediksi di masa mendatang atau dengan kata lain mengajarkan dan melatih model dengan beberapa data dari dataset yang berlabel. Terdapat dua tipe *supervised learning* diantaranya, klasifikasi dan regresi. (DigitalTalentScholarship, 2019)

b. *Unsupervised Learning*

Unsupervised Learning adalah pembelajaran tanpa pengawasan, maksudnya membiarkan model bekerja sendiri untuk menemukan informasi yang mungkin tidak terlihat oleh mata manusia. Algoritma *unsupervised learning* melatih dataset, dan menarik kesimpulan pada data tidak berlabel (*unlabeled*). Teknik *unsupervised learning* diantaranya adalah *dimension reduction* (reduksi dimensi), *density estimation* (estimasi kepadatan), *market basket analysis* (analisis keranjang belanja) dan *clustering* (pengelompokan) (DigitalTalentScholarship, 2019).

c. *Reinforcement Learning*

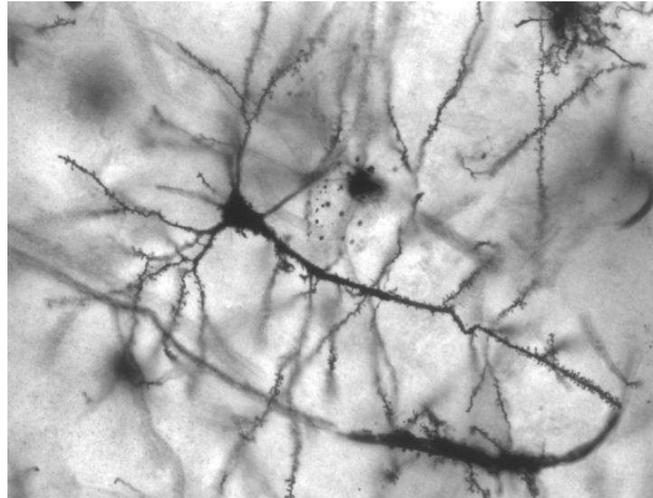
Herlambang (2020) mengungkapkan bahwa *reinforcement learning* adalah teknik dimana model belajar dari pengalaman yang didapat layaknya manusia yang juga belajar dari pengalaman. Semakin lama usia manusia tersebut, semakin banyak pula pengalamannya. Contohnya, ketika manusia mengajarkan robot caranya berjalan, berlari, berbicara dan aktivitas lain layaknya manusia.

Dalam penelitian ini, kegunaan *machine learning* mengacu pada poin pertama, yaitu *supervised learning* khususnya klasifikasi. Klasifikasi adalah proses memprediksi label atau kategori kelas diskrit.

2.2 Deep Learning

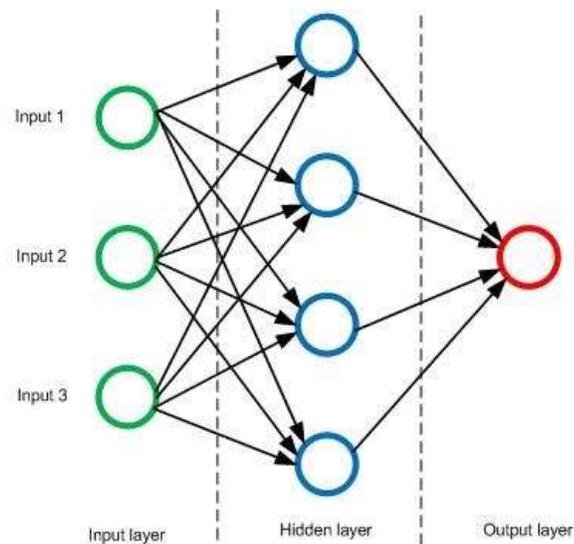
Deep learning dipopulerkan oleh Geoffrey Hinton. Pada tahun 2006 ia memperkenalkan *neural network* (jaringan saraf tiruan) dan mempopulerkannya dengan istilah *deep learning*. Menurut Maulana & Rochmawati (2019) *deep learning* merupakan proses pembelajaran yang menggunakan algoritma yang mengacu pada hukum matematik yang bekerja seperti otak pada manusia. Deng dan Yu (2013 dalam Arrofiqoh & Harintaka, 2018) juga menyatakan bahwa *deep learning* adalah cabang dari *machine learning* yang dapat mengajarkan komputer untuk melakukan pekerjaan selayaknya manusia, seperti komputer dapat belajar dari proses *training*. Berbagai macam pekerjaan memanfaatkan *deep learning*, contohnya memprediksi peluang atau kejadian, mengenali objek, hingga mendiagnosa penyakit. Selain itu, *deep learning* dapat melakukan ekstraksi fitur, pengenalan pola dan klasifikasi.

Bentuk *deep learning* sangat mirip dengan saraf di otak manusia. Berikut adalah ilustrasi saraf (neuron) otak manusia.



Gambar 2. 1 Ilustrasi Saraf pada Otak Manusia

Berdasarkan ilustrasi diatas, neuron tampak seperti benang-benang yang saling terhubung. Pada bagian intinya (nucleus) terlihat besar dan saling terhubung dengan inti saraf lainnya melalui dendrit dan akson. Jumlah saraf pada otak manusia tidak hanya satu melainkan jutaan bahkan sampai milyaran saraf neuron. Berikut adalah contoh ilustrasi beberapa lapis neuron dalam *deep learning*.



Gambar 2. 2 Ilustrasi *Deep Learning* dengan 1 *Hidden Layer*

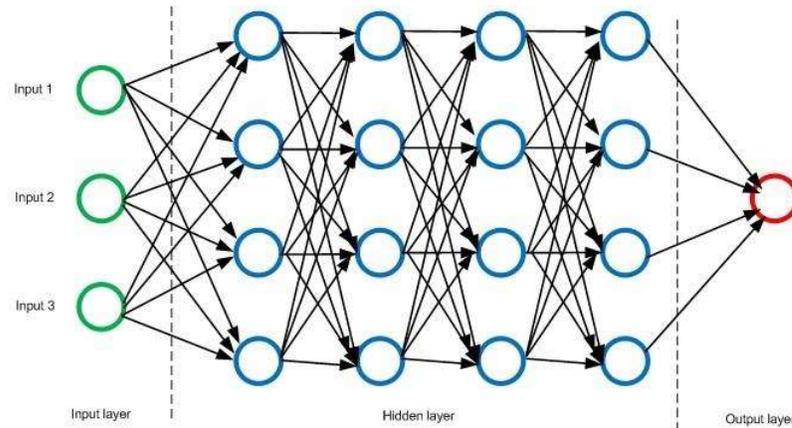
Melalui ilustrasi diatas, dapat dilihat struktur sederhana *deep learning*. Struktur ini adalah sistem saraf tiruan yang dimodelkan dari saraf otak manusia.

Terdapat tiga bagian/*layer* pada *deep learning*, diantaranya *input layer*, *hidden layer* dan *output layer*.

- *Input layer* adalah neuron yang menerima masukan (input). Dalam ilustrasi di atas ada 3 neuron yang menerima input. Satu neuron di dalam arsitektur *neural networks* sering disebut dengan istilah *node*.
- *Output layer* adalah nilai akhir (keputusan akhir/ kondisi akhir) yang ingin dihasilkan dari nilai-nilai awal yang dimasukkan di *input layer*. Jika kita ingin memprediksi harga rumah, maka *output layer* adalah nilai harga rumah yang dihasilkan berdasarkan beberapa nilai input yang sudah dimasukkan.
- *Hidden layer* merupakan neuron yang menjadi perantara antara *input layer* dengan *output layer*. Layaknya mata yang melihat sebuah objek, maka bisa dianalogikan bahwa mata adalah *input layer*, otak adalah *output layer* (tujuan akhir untuk memproses input objek oleh mata), dan *hidden layer* adalah sebuah neuron yang dilewati dari mata sampai ke otak.

Jumlah neuron di dalam sebuah *hidden layer* sangat mempengaruhi performa *deep learning*. Jika neuron (nodes) nya terlalu sedikit maka outputnya bisa *underfitting*. Jika neuronnya terlalu banyak maka bisa menjadi *overfitting*.

Ilustrasi diatas hanya ada satu *hidden layer*, di dalam *deep learning*, jumlah *hidden layer* tidak hanya satu, bisa dua, tiga bahkan lebih selama *hardware* nya kuat untuk mengakomodir komputasi yang ada. Sebuah arsitektur *deep learning* yang terdiri dari beberapa *hidden layer* tampak sebagai berikut



Gambar 2. 3 Ilustrasi *Deep Learning* dengan beberapa *Hidden Layer*

Untuk merangkum pembahasan *deep learning*, bisa dikatakan bahwa *deep learning* adalah sebuah teknik *machine learning* di mana proses pengambilan keputusan dilakukan melalui beberapa sistem saraf tiruan manusia. Adanya teknik *deep learning*, performa sebuah program AI sangatlah baik, cepat dan lebih akurat, jika dibandingkan dengan teknik AI atau *machine learning* konvensional

Bidang yang memanfaatkan *deep learning* salah satunya *image processing* atau pengolahan citra digital. Sistem *image processing* berfungsi untuk membantu manusia dalam mengenali atau mengklasifikasi objek dengan efisien yaitu cepat, tepat dan dapat memproses banyak data sekaligus (Maulana & Rochmawati, 2019).

Berikut merupakan contoh algoritma *deep learning* beserta contoh aplikasinya.

	Nama Teknik DL	Aplikasi
Supervised	<i>Artificial Neural Networks</i>	Regresi dan klasifikasi
	<i>Convolutional Neural Networks</i>	<i>Computer vision</i>
	<i>Recurrent Neural Networks</i>	Analisis data kontinu (<i>time-series</i>)
Unsupervised	<i>Self-Organizing Maps</i>	<i>Feature detection</i>
	<i>Deep Boltzmann Machines</i>	Untuk rekomendasi sistem
	<i>Auto Encoders</i>	Untuk rekomendasi sistem

Gambar 2. 4 Macam-macam Algoritma *Deep Learning*

Penelitian ini menerapkan teknik *deep learning supervised* karena objek penelitian akan dilakukan klasifikasi. Selain itu, karena objek penelitiannya merupakan sebuah citra, maka digunakan algoritma CNN.

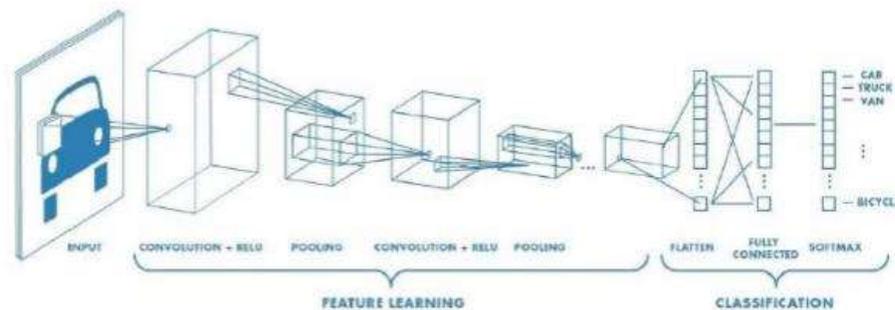
2.3 Convolutional Neural Network (CNN)

Suartika et al (2016 dalam Pangestu & Bunyamin, 2018) menyatakan bahwa CNN adalah pengembangan dari *Multilayer Perceptron* (MLP) untuk mengolah data dua dimensi. *Multilayer Perceptron* sendiri adalah pengembangan dari *Artificial Neural Network* (ANN) yang ditujukan untuk menutupi kekurangan *Artificial Neural Network* (ANN) dengan *Single-layer Perceptron* dalam menyelesaikan operasi logika yang kompleks (Krisnadi et al, 2008 dalam Pangestu & Bunyamin, 2018). Hal tersebut dimungkinkan dengan menambahkan *hidden layer* yang membuat ANN powerful untuk memecahkan operasi logika yang kompleks (*universal approximation*) (Alpaydin, 2010 dalam Pangestu & Bunyamin, 2018) dan sering digunakan untuk permasalahan-permasalahan klasifikasi, recognition, dan prediksi (Pangestu & Bunyamin, 2018).

Terdapat beberapa algoritma yang digunakan pada bidang *image processing* diantaranya, *Naive Bayes*, *Support Vector Machine* dan *Neural Network*. Algoritma *Neural Network* yang telah dikembangkan menjadi CNN ini memiliki hasil yang paling signifikan dalam pengenalan citra digital. Hal tersebut dikarenakan CNN diimplementasikan berdasarkan sistem pengenalan citra pada visual cortex manusia (Maulana & Rochmawati, 2019). Hal ini juga diperkuat oleh Dewa (2018 dalam Lauw et al., 2019) yang mengungkapkan bahwa kemampuan CNN di klaim sebagai

model terbaik untuk memecahkan permasalahan *object detection* dan *object recognition*.

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (input) dan keluaran (output) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. *Convolutional neural network* sendiri adalah perpaduan antara konvolusi citra untuk proses ekstraksi fitur, dan *neural network* untuk klasifikasi. Berikut adalah jaringan arsitektur *Convolutional Neural Network* (Arifin, 2019):



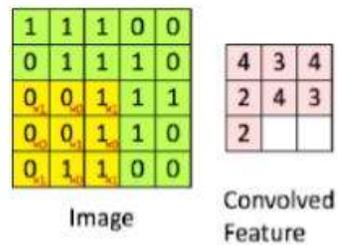
Gambar 2. 5 Arsitektur CNN

Berdasarkan arsitektur LeNet5 (Stanford, 2016), terdapat 4 macam *layer* utama pada sebuah CNN yaitu *Convolutional layer*, *relu layer*, *subsampling layer/pooling layer*, dan *fully connected layer*. Berikut ini adalah penjelasan mengenai masing-masing *layer* (Arifin, 2019).

a. *Convolution Layer*

Convolutional Layer melakukan operasi konvolusi pada output dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra,

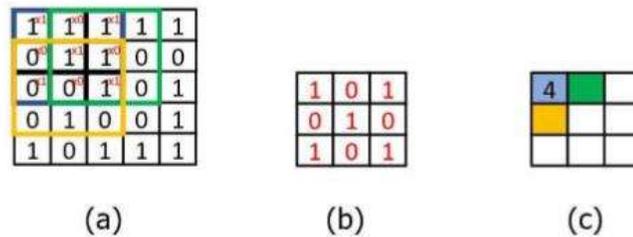
konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra disemua offset yang memungkinkan seperti yang ditunjukkan pada Gambar 2.6 Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah (Arifin, 2019).



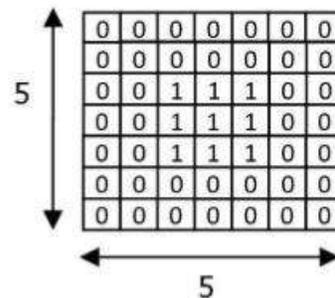
Gambar 2. 6 Operasi Konvolusi

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input menggunakan filter. Filter ini berisi bobot yang digunakan untuk mendeteksi karakter dari objek seperti tepi, kurva, atau warna. Konvolusi akan menghasilkan transformasi linear dari citra input yang sesuai dengan informasi spasial pada data. Bobot pada *layer* tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN. Filter diaplikasikan secara berulang sehingga menghasilkan serangkaian bidang *receptive*. Terdapat parameter yang dapat diubah untuk memodifikasi sifat tiap lapisan, yaitu ukuran filter, stride dan padding. *Stride* mengontrol bagaimana filter diterapkan pada data input dengan bergerak sepanjang ukuran piksel yang telah ditentukan. *Padding* adalah penambahan ukuran piksel dengan nilai tertentu disekitar data input agar hasil dari bidang *receptive* tidak terlalu kecil sehingga tidak banyak informasi yang hilang. Nilai ini biasanya nol sehingga disebut dengan zero padding. Hasil

dari bidang receptive berupa data tunggal. *Output* dari proses konvolusi ini dijadikan sebagai input untuk lapisan konvolusi selanjutnya (Castelluccio et al., 2015 dalam Arrofiqoh & Harintaka, 2018). Ilustrasi proses *stride* dan *padding* ditampilkan pada Gambar 2.7 dan Gambar 2.8.



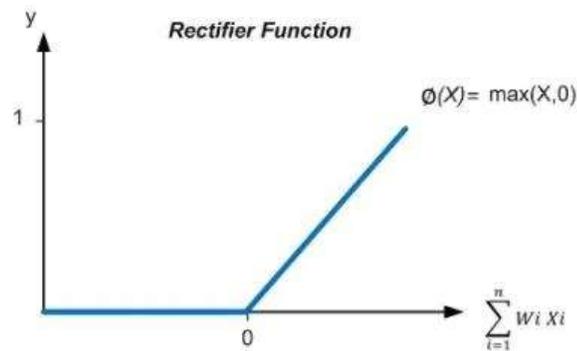
Gambar 2. 7 Operasi konvolusi dengan stride 1 (a) Input data 5x5 (b) filter 3x3 (c) bidang receptive 3x3.



Gambar 2. 8 Zero Padding 2 pada data 3x3.

b. ReLu Layer (Fungsi Aktivasi ReLU)

ReLU (*Rectification Linear Unit*) merupakan operasi untuk mengenalkan nonlinearitas dan meningkatkan representasi dari model. Fungsi aktivasi ReLU adalah $f(x) = \max(0, x)$ (Heaton, 2015 dalam Nour, 2018). Nilai output dari neuron bisa dinyatakan sebagai 0 jika inputnya adalah negatif. Jika nilai input adalah positif, maka output dari neuron adalah nilai input aktivasi itu sendiri (Kim et al., 2016 dalam Arrofiqoh & Harintaka, 2018).

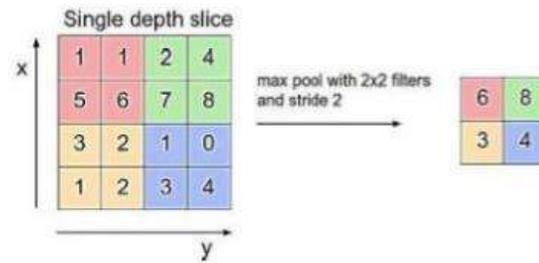


Gambar 2. 9 Fungsi Aktivasi ReLU

c. *Pooling (Subsampling) Layer*

Pooling atau *subsampling* adalah proses mereduksi ukuran sebuah data citra atau dengan kata lain pengurangan ukuran matriks. Dalam pengolahan citra, *subsampling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Menurut Bejiga et al. (2017) terdapat dua macam *pooling* yang sering digunakan yaitu *average pooling* dan *max pooling*. Zhi et al. (2016) mengatakan nilai yang diambil pada *average pooling* adalah nilai rata-rata sedangkan pada *max pooling* adalah nilai maksimal (Arrofiqoh & Harintaka, 2018).

Dalam sebagian besar CNN, metode subsampling yang digunakan adalah *maxpooling*. *Max pooling* membagi output dari *Convolutional layer* menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.10 Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran) (Arifin, 2019).

Gambar 2. 10 Operasi Max *Pooling*

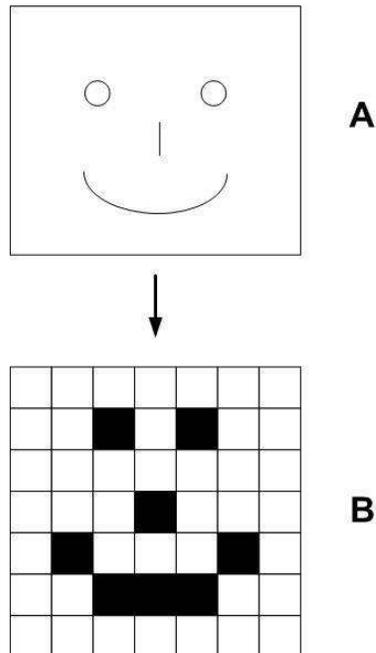
Menurut Springenberg (J. T. Springenberg, 2015), penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *Convolutional layer* dengan stride yang sama dengan *pooling layer* yang bersangkutan.

d. *Fully Connected Layer*

Lapisan *fully connected layer* merupakan kumpulan dari proses konvolusi (Hijazi et al., 2015). Lapisan ini mendapatkan input dari proses sebelumnya untuk menentukan fitur mana yang paling berkorelasi dengan kelas tertentu. Fungsi dari lapisan ini adalah untuk menyatukan semua node menjadi satu dimensi (Albelwi & Mahmood, 2017 dalam Arrofiqoh & Harintaka, 2018).

2.4 Cara Kerja *Convolutional Neural Network (CNN)*

Untuk dapat lebih memahami cara kerja dari proses konvolusi, dapat dilihat ilustrasi berikut.



Gambar 2. 11 Gambar Sebelum dilakukan Konvolusi

Pada Gambar 2.11 adalah gambar yang berukuran 7 x 7. Peneliti akan mengubah gambar diatas menjadi sebuah array yang berukuran 7 x 7.

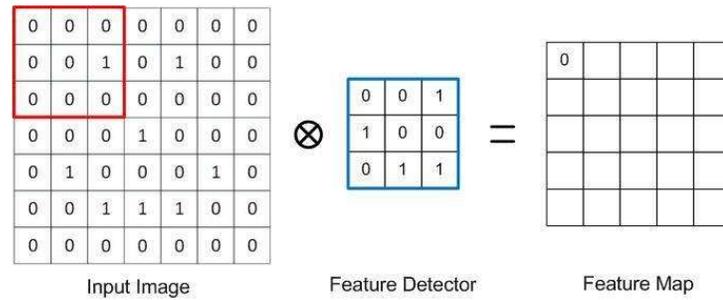
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Gambar 2. 12 Gambar Berukuran 7 x 7 dalam Bentuk Array

Terdapat 4 tahapan CNN, diantaranya:

1. *Convolution*

Sesuai namanya, CNN (*Convolutional Neural Networks*) terinspirasi oleh langkah pertamanya, yaitu *convolution*. Pada ilustrasi diatas, peneliti akan menggunakan kernel/*feature detector* yang berukuran 3x3 untuk konvolusi.



Gambar 2. 13 Alur Mendapatkan Feature Map (i)

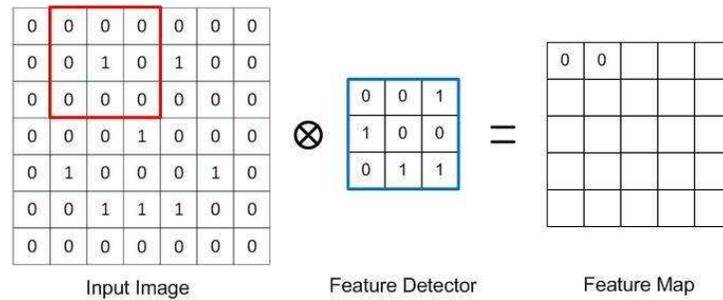
Gambar sebelah kiri merupakan gambar senyum (Gambar 3.1) yang sudah dirubah kedalam bentuk array dengan ukuran 7x7. Nilai pixel untuk gambar ini hanya 0 dan 1 untuk mempermudah pemahaman. Adapun kernel atau feature detector yang berukuran 3x3 yang akan digunakan untuk konvolusi dapat dilihat pada Gambar 2.13.

Dengan menggunakan kernel 3x3 dengan strided atau langkah yang digunakan dalam perhitungan konvolusi tersebut adalah 1 maka proses perhitungan konvolusi tersebut dapat divisualisasikan seperti pada Gambar 2.14.

Kemudian yang terjadi dalam tahap *convolution* adalah mengalikan pixel pada gambar (ditandai dengan warna merah) dengan *feature detector*. Hasil dari perkalian tiap pixelnya kemudian dijumlahkan. Ilustrasinya sebagai berikut:

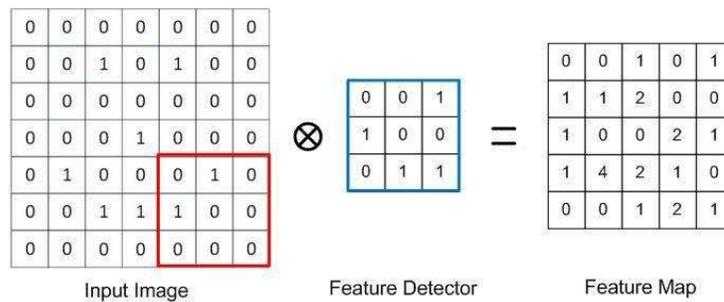
$$(0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1)$$

$$= 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0$$



Gambar 2. 14 Alur Mendapatkan Feature Map (ii)

Proses konvolusi berlangsung sampai sudut kanan paling bawah. Setelah proses perhitungan akan menghasilkan sebuah array baru yang berukuran 5x5.



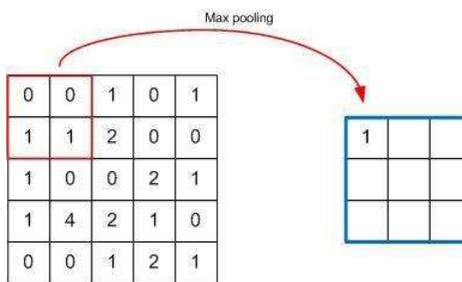
Gambar 2. 15 Alur Mendapatkan Feature Map (iii)

Dapat dilihat pada proses konvolusi diatas bahwa gambar awal yang berukuran 7x7 berkurang menjadi lebih kecil yaitu 5x5. Hal tersebut merupakan salah satu tujuan utama *feature detector*. Melalui *feature detector* proses pengolahan gambar menjadi semakin cepat karena data pixel yang diolah juga semakin kecil.

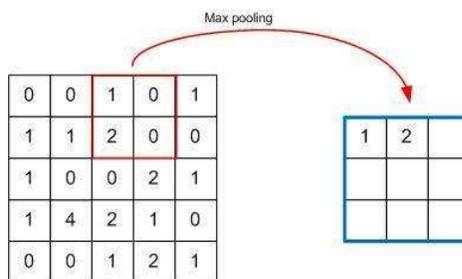
Setelah hasil konvolusi didapatkan, langkah selanjutnya adalah menggunakan *Rectifier Linear Units (ReLU) activation function* untuk mengurangi linearitas dari *feature map* yang ada di *convolutional layer*. Dengan menggunakan ReLU, maka nilai negatif pada pixel akan dikonversi menjadi nol.

2. Max Pooling

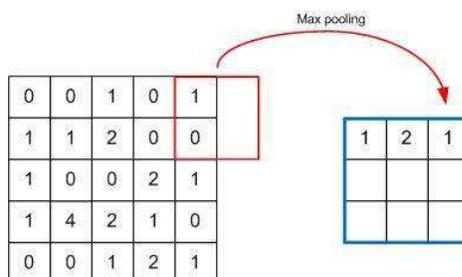
Setelah hasil konvolusi didapatkan, maka masuk ke proses selanjutnya yaitu proses *maxpooling*. Proses *maxpooling* hampir sama dengan proses konvolusi. Proses *maxpooling* akan mengambil nilai tertinggi dari suatu input citra. Dalam penerapan pada sistem ini, proses *maxpooling* akan menggunakan pixel 2x2. Setiap 2x2 pixel pada Input citra akan diambil nilai pixel tertingginya. Pengambilan nilai tertinggi pada pixel akan terus bergeser hingga seluru pixel pada suatu citra telah habis. Untuk lebih jelasnya, ilustrasi akan ditampilkan pada Gambar 2.16 sampai dengan Gambar 2.19.



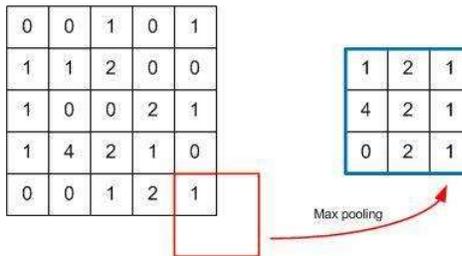
Gambar 2. 16 Proses *Maxpooling* (i)



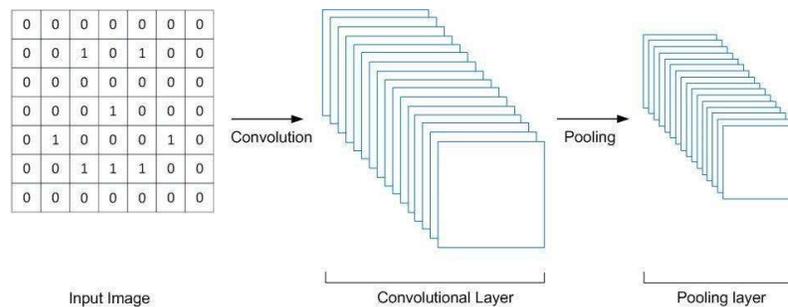
Gambar 2. 17 Proses *Maxpooling* (ii)



Gambar 2. 18 Proses *Maxpooling* (iii)

Gambar 2. 19 Proses Akhir *Maxpooling*

Dari proses *pooling* ini akan dihasilkan *pooling layer* (*pooled feature maps*). Tentunya dalam CNN kita melakukan beberapa kali proses *pooling* sehingga ada banyak *pooling layer* yang dihasilkan. Tampilannya adalah sebagai berikut.

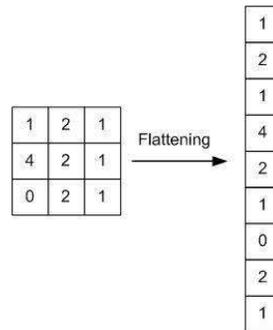
Gambar 2. 20 Proses Awal Hingga *Pooling Layer*

Pooling layer digunakan untuk mengurangi ukuran dan mempercepat perhitungan, serta membuat beberapa fitur yang diperkirakan sedikit lebih akurat (Priyono 2018 dalam Ilahiyah & Nilogiri, 2018).

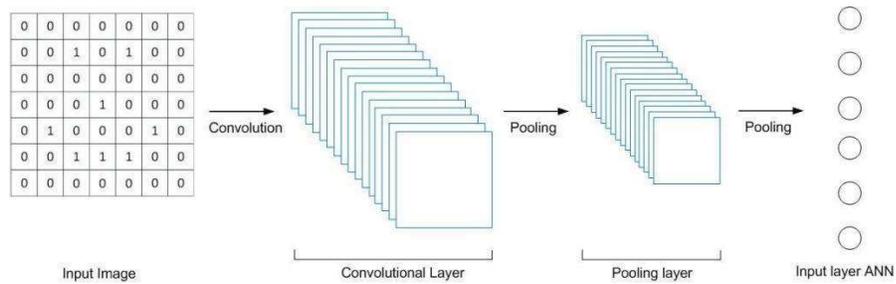
3. *Flattening*

Setelah proses *maxpooling* ini selesai, maka masuk ke proses selanjutnya. Yaitu proses *flattening*. Proses ini merupakan proses terpenting karena proses ini akan menghasilkan nilai yang akan digunakan untuk proses *training* dengan *neural network*. Proses ini berguna untuk mempermudah komputasi dengan cara merubah array yang berdimensi $n \times m$ menjadi 1 deret bilangan sebagai Vektor Input. Mengacu pada contoh sebelumnya, hasil

maxpooling 3 x 3 akan berubah menjadi 9 x 1. Untuk memperjelas hasil *flattening* penulis akan menampilkan hasilnya pada Gambar 2.21.



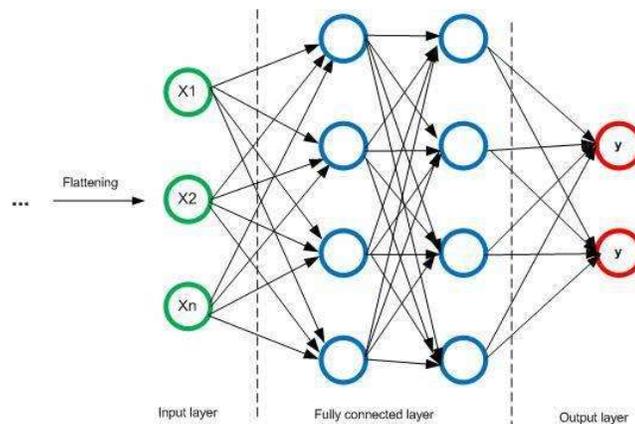
Gambar 2. 21 Hasil *Flattening*



Gambar 2. 22 Proses Awal Hingga *Flattening*

4. *Fully Connected*

Tahap selanjutnya yaitu memasukan hasil *flattening* kedalam struktur *neural network* yang utuh sehingga akan diketahui output yaitu klasifikasi hama pada tanaman padi.



Gambar 2. 23 Ilustrasi *Fully Connected*

2.4 Python

Python adalah bahasa pemrograman model skrip (*scripting language*) yang berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya (Triasanti, 2001 dalam Arifin, 2019).

2.5 Tensorflow

Framework Tensorflow dikembangkan oleh Google Brain Team pada tahun 2015 untuk perhitungan numerik, dan sekarang banyak digunakan oleh perusahaan besar untuk pengembangan aplikasi AI, seperti klasifikasi citra, penyematan kata, dan pengembangan *chatbot* (Padilha&Lucena dalam Wiranda et al., 2020). *Library tensorflow* ini merupakan *interface* untuk mengekspresikan algoritma machine learning dan untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki tentang objek tersebut atau target yang dikenali serta dapat membedakan objek satu dengan objek lainnya. *Tensorflow* memiliki fitur untuk menjalankan pelatihan model menggunakan *Central Processing Unit* (CPU) dan pelatihan model *Graphic Processing Unit* (GPU) (Nurfita & Ariyanto, 2018).

2.6 Tanaman Padi

Tanaman padi (*Oryza sativa* L.) merupakan tanaman pangan penting yang menjadi makanan pokok lebih dari setengah penduduk dunia karena mengandung nutrisi yang diperlukan tubuh. Menurut Poedjiadi (1994), kandungan karbohidrat padi giling sebesar 78,9 %, protein 6,8 %, lemak 0,7 % dan lain-lain 0,6 %. Indonesia sebagai negara dengan jumlah penduduk yang besar menghadapi tantangan dalam memenuhi kebutuhan pangan tersebut (Pratiwi, 2016).



Gambar 2. 24 Tanaman Padi

Menurut Wati (2017) padi merupakan tanaman serealialia penting dan digunakan sebagai makanan pokok oleh bangsa Indonesia. Itulah sebabnya produksi padi sangat perlu untuk ditingkatkan. Peningkatan produksi padi dipengaruhi faktor pengganggu yang dapat berakibat pada penurunan produksi. Beberapa faktor yang mempengaruhi tinggi rendahnya produksi padi adalah penggunaan varietas, pemakaian pupuk, cara bercocok tanam, serta jasad pengganggu (OPT).

Kendala utama yang sering dihadapi oleh petani adalah adanya Organisme Pengganggu Tanaman (OPT). Dimana Organisme pengganggu ini berupa hama,

penyakit dan gulma yang dapat menyebabkan rendahnya produktivitas padi per hektar, bahkan dapat menyebabkan gagal panen. Identifikasi hama-hama yang ada di lapangan perlu untuk diketahui agar dapat diketahui jenis, tingkat serangan dan cara pengendalian yang tepat dilakukan sesuai dengan sasaran.

2.7 Hama

Hama dalam arti luas adalah semua bentuk gangguan baik pada manusia, ternak dan tanaman. Pengertian hama dalam arti sempit yang berkaitan dengan kegiatan budidaya tanaman adalah semua hewan yang merusak tanaman atau hasilnya yang mana aktivitas hidupnya ini dapat menimbulkan kerugian secara ekonomis. Adanya suatu hewan dalam satu pertanaman sebelum menimbulkan kerugian secara ekonomis maka dalam pengertian ini belum termasuk hama. Namun demikian potensi mereka sebagai hama nantinya perlu dimonitor dalam suatu kegiatan yang disebut pemantauan (monitoring). Secara garis besar hewan yang dapat menjadi hama dapat dari jenis serangga, moluska, tungau, tikus, burung, atau mamalia besar (Sulistianingsih et al., 2014).

2.8 State of The Art

No	Konten	Deskripsi
1	Paper ke-1	
	Judul Paper	Klasifikasi Spesies Lebah Berbasis Data Citra dengan Metode <i>Support Vector Machine</i>
	Penulis	Ariyadi
	Jurnal/Konferensi	Jurnal Inovasi Penelitian Vol.1 No. 6 Nopember 2020 ISSN 2722-9475 (Cetak) ISSN 2722-9467 (Online)
	URL	https://stp-mataram.e-journal.id/JIP/article/view/204
	Permasalahan	Lebah bumble dan lebah madu sering kali membingungkan karena kedua spesies tersebut memiliki kemiripan. Penurunan jumlah lebah pekerja menyebabkan peningkatan biaya petani hingga 20 % dalam produksi madu karena harus menyewa jasa penyerbukan. Identifikasi beragam spesies lebah di alam liar membuat pemahaman tentang penyebaran dan pertumbuhan serangga ini.
	Kontribusi	Untuk memaksimalkan proses komputasi, fitur data telah diolah dengan Histogram of Oriented Gradient (HOG) dan Principal Component Analysis (PCA).
	Metode/Solusi	Klasifikasi lebah madu atau lebah bumble berbasis citra digital menggunakan <i>Support Vector Machine</i> (SVM)
	Hasil Utama	Hasil akhir penelitian ini menunjukkan bahwa model yang telah dibangun dengan metode SVM menghasilkan tingkat akurasi 74%.
Batasan	Hanya dua spesies lebah yang diambil untuk proses klasifikasi, yaitu lebah madu dan lebah bumble	
2	Paper ke-2	
	Judul Paper	Rancang Bangun Aplikasi Klasifikasi Jenis Kupu-Kupu Awetan Family Papilionidae Dengan Metode SVM
	Penulis	Asslia Johar Latipah, Naufal Azmi Verdhika, Siti Puspita Hida Sakti MZ

	Jurnal/Konferensi	JURTI, Vol.2 No.2, Desember 2018, ISSN: 2579-8790
	URL	https://www.researchgate.net/publication/349528890_Rancang_Bangun_Aplikasi_Klasifikasi_Jenis_Kupu-Kupu_Awetan_Family_Papilionidae_Dengan_Metode_SVM
	Permasalahan	Kupu-kupu family Papilionidae dapat dijadikan bahan praktikum taksonomi karena sayapnya yang lebar dan kokoh sehingga bentuk venasi terlihat jelas. Dengan adanya aplikasi pengenalan jenis kupu-kupu maka akan menghemat biaya dokumentasi dan mampu membantu dalam mengenali jenis kupu-kupu awetan.
	Kontribusi	Algoritma klasifikasi yang digunakan di dalam penelitian ini adalah Support Vector Machine Linear dengan kernel Linear.
	Metode/Solusi	Membuat aplikasi dengan sistem untuk mengenali jenis dari suatu spesies kupu-kupu dari kelas Papilionidae. Algoritma klasifikasi yang digunakan di dalam penelitian ini adalah Support Vector Machine Linear dengan kernel Linear.
	Hasil Utama	Terdapat kesalahan dalam klasifikasi yang dilakukan yaitu 3 kesalahan dari 20 kali pengujian
	Batasan	Kupu-kupu yang menjadi objek penelitian adalah kupu-kupu dari famili Papilionidae Wilayah yang akan di klasifikasi (Region of Interest) yaitu sayap kupu-kupu sebelah kanan.
3	Paper ke-3	
	Judul Paper	Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network
	Penulis	Sarirotul Ilahiyah & Agung Nilogiri
	Jurnal/Konferensi	JUSTINDO (Jurnal Sistem & Teknologi Informasi Indonesia), Vol. 3, No. 2, Agustus 2018, p-ISSN : 2502-5724; e-ISSN : 2541-5735
	URL	http://jurnal.unmuhjember.ac.id/index.php/JUSTINDO/article/view/2254

	Permasalahan	Klasifikasi jenis genus tumbuhan berdasarkan citra daun pada berbagai macam posisi yang mungkin.
	Kontribusi	digunakan arsitektur CNN dari Krizhevsky et al.(2012) yang disebut dengan AlexNet
	Metode/Solusi	Proses klasifikasi dilakukan dengan metode CNN yang dibuat dengan prinsip <i>translation invariance</i>
	Hasil Utama	Sistem yang telah dibuat dapat mengidentifikasi jenis genus tumbuhan dengan nilai akurasi sistem sebesar 90,8%.
	Batasan	20 jenis genus tumbuhan yang diklasifikasikan dengan masing-masing memiliki 100 citra daun berbagai posisi
4	Paper ke-4	
	Judul Paper	Implementasi Metode Support Vector Machine untuk Identifikasi Penyakit Daun Tanaman Kubis
	Penulis	Ariadi Retno Tri Hayati Ririd, Ayundha Wulan Kurniawati, Yoppy Yunhasnawa
	Jurnal/Konferensi	Jurnal Informatika Polinema Volume 4, Edisi 3, Mei 2018, e-ISSN: 2407-070X p-ISSN: 2614-6371
	URL	https://www.researchgate.net/publication/332044428_IMPLEMENTASI_METODE_SUPPORT_VECTOR_MACHINE_UNTUK_IDENTIFIKASI_PENYAKIT_DAUN_TANAMAN_KUBIS
	Permasalahan	Bercak daun dan busuk hitam merupakan penyakit yang menyerang daun tanaman kubis dan menimbulkan kerugian sehingga produksi kubis mengalami penurunan. Identifikasi penyakit daun tanaman kubis dilakukan secara manual dengan mengamati gejala yang tampak pada daun tapi tidak semua petani memiliki pemahaman detail mengenai penyakit yang menyerang tanaman kubis.
	Kontribusi	Pengaruh parameter pada Sequential Training terhadap nilai akurasi sistem.

	Metode/Solusi	Proses identifikasi penyakit daun tanaman kubis menggunakan metode SVM dengan menggunakan Sequential Training dengan menghitung nilai yang didapatkan dari preprocessing.
	Hasil Utama	Rata-rata akurasi hasil klasifikasi mencapai 80.55%.
	Batasan	<ol style="list-style-type: none"> 1. Parameter-parameter yang digunakan pada Sequential Training adalah Lambda, Gamma, C, Sigma, Epsilon dan Maximal Iteration. 2. Penyakit daun tanaman kubis hanya diambil dua jenis, yaitu bercak daun dan busuk hitam.
5	Paper ke-5	
	Judul Paper	<i>Insect Classification and Detection in Field crops using modern machine learning techniques</i>
	Penulis	Thenmozhi Kasinathan, Dakshayani Singaraju, Srinivasulu Reddy Uyyala
	Jurnal/Konferensi	<i>Information Processing in Agriculture</i> , 26 Juni 2020
	URL	https://www.sciencedirect.com/science/article/pii/S2214317320302067
	Permasalahan	Serangga adalah alasan utama dibalik penurunan kualitas tanaman dan oleh karena itu menurunkan produktivitas tanaman. Identifikasi serangga tradisional memiliki kelemahan yaitu membutuhkan ahli taksonomi yang terlatih untuk mengidentifikasi serangga berdasarkan ciri morfologi secara akurat.
	Kontribusi	Validasi silang 9 kali lipat diterapkan untuk meningkatkan kinerja model klasifikasi
	Metode/Solusi	Mengklasifikasikan dan mendeteksi serangga pada jagung, kedelai, gandum, dll dengan menggunakan machine learning dan algoritma deteksi hama serangga pada tahap awal pertumbuhan tanaman (JST, SVM, KNN, NB dan CNN).
	Hasil Utama	Tingkat klasifikasi tertinggi 91,5% dan 90% dicapai untuk 9 dan 24 kelas serangga menggunakan model CNN.

	Batasan	Validasi silang 9 kali lipat diterapkan untuk meningkatkan kinerja model klasifikasi.
6	Paper ke-6	
	Judul Paper	Klasifikasi Hama dan Penyakit Tanaman Jagung dengan Menggunakan Neural Network Berbasis Algoritma Genetika
	Penulis	Rais
	Jurnal/Konferensi	SENIT 2016 ISBN: 978-602-74355-0-6 51
	URL	https://ejournal.poltektegal.ac.id/index.php/prosiding/article/view/357
	Permasalahan	Kerusakan akibat hama penyakit tanaman jagung menyebabkan kerugian hasil panen bagi para petani. Pengelompokan hama penyakit tanaman jagung sangatlah penting bagi para petani agar dengan mudah petani mengenal hama penyakit tanaman jagung
	Kontribusi	Algoritma genetika digunakan untuk optimasi parameter Neural Network seperti jumlah hidden layer dan learning rate agar akurasi yang dihasilkan bisa lebih bagus.
	Metode/Solusi	Penggunaan Neural Network (NN) Backpropagation untuk klasifikasi hama penyakit tanaman jagung. Untuk meningkatkan tingkat akurasi klasifikasi hama penyakit tanaman jagung maka menerapkan algoritma genetika (GA).
	Hasil Utama	Untuk model Neural Network, nilai accuracy yang didapatkan sebesar 96.60%, setelah dioptimasi nilai akurasi algoritma Neural Network berbasis Algoritma Genetika nilai accuracy yang didapatkan sebesar 97.20%.
	Batasan	Dalam menentukan jumlah training cycles dilakukan beberapa percobaan untuk mendapatkan jumlah training cycles terbaik, pada eksperimen ini digunakan nilai training cycles dari 100 – 1000.
7	Paper ke-7	

	Judul Paper	Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan Convolutional Neural Network Berbasis Android
	Penulis	Kevin Oktovio Lauw, Leo Willyanto Santoso, Rolly Intan
	Jurnal/Konferensi	Jurnal Infra Vol 8, No 2 (2020)
	URL	http://publication.petra.ac.id/index.php/teknik-informatika/article/view/10496
	Permasalahan	Untuk memelihara seekor anjing, terdapat beberapa faktor yang harus diperhatikan seperti pakan yang dikonsumsi, intensitas perawatan, serta kebersihan kandang atau lingkungan yang sesuai.
	Kontribusi	penelitian ini menggunakan YOLO untuk mengidentifikasi objek dalam gambar yaitu anjing
	Metode/Solusi	YOLO (You Only Look Once) untuk mengidentifikasi objek dalam gambar yaitu anjing setelah itu, gambar tersebut akan diproses oleh CNN untuk mengidentifikasi jenis anjing melalui gambar yang diberikan dan memberikan informasi yang berkaitan seperti nama jenisnya, sifatnya, dan cara untuk memelihara anjing tersebut.
	Hasil Utama	Akurasi dari YOLO untuk mendeteksi anjing sebesar 94,242%, akurasi CNN model I sebesar 56,400%, akurasi CNN model II sebesar 40,000% dan akurasi CNN model III 50,400%.
	Batasan	Pengujian dengan menggunakan metode YOLO dilakukan terhadap 25 class anjing yang berbeda
8	Paper ke-8	
	Judul Paper	Klasifikasi Penyakit Tanaman Apel dari Citra Daun Dengan Convolutional Neural Network
	Penulis	Andi Asrafil Ardan Paliwang, M Ridwan Dwi Septian, Margi Cahyanti, dan Ericks Rachmat Swedia
	Jurnal/Konferensi	SEBATIK 1410-3737

	URL	https://jurnal.wicida.ac.id/index.php/sebatik/article/view/1060
	Permasalahan	Terkadang petani juga membutuhkan seorang pakar dalam menentukan jenis hama dan penyakit pada tanaman Apel agar dapat memberikan solusi terbaik.
	Kontribusi	Aplikasi ini dibuat menggunakan Flutter dan Visual Studio Code dan dapat dijalankan pada smartphone dengan minimum versi Android Lollipop.
	Metode/Solusi	Penerapan metode CNN dalam klasifikasi penyakit tanaman apel dengan menggunakan bahasa pemrograman Python 3.6 dan Google Colab.
	Hasil Utama	Hasil dari final test accuracy yang diperoleh dari keseluruhan proses pelatihan yaitu sebesar 97,1%.
	Batasan	<ol style="list-style-type: none"> 1. Penyakit yang digunakan untuk klasifikasi hanya 3 penyakit, yaitu scab, blackrot, dan rust. 2. Aplikasi ini hanya dapat dijalankan pada smartphone yang memiliki sistem operasi Android. 3. Output berupa hasil klasifikasi jenis penyakit tanaman Apel.
9	Paper ke-9	
	Judul Paper	Implementasi Convolutional Neural Network Untuk Identifikasi Penyakit Daun Gambas
	Penulis	Dwi Fitriana Sari, Daniel Swanjaya
	Jurnal/Konferensi	Seminar Nasional Inovasi Teknologi Kediri, 25 Juli 2020 e-ISSN: 2549-7952 UN PGRI p-ISSN: 2580-3336
	URL	https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/76
	Permasalahan	Kurangnya pengetahuan tentang hama dan penyakit yang menyerang tanaman gambas bisa mengakibatkan kegagalan panen karena proses penanganan yang kurang tepat sehingga tanaman Gambas menjadi tidak sehat dan mudah mati.
	Kontribusi	Menerapkan arsitektur MobileNet dengan kedalaman layer 28 dan jenis pooling Average Pooling

	Metode/Solusi	Penelitian menggunakan Convolutional Neural Network (CNN) bertujuan untuk menerapkan model arsitektur MobileNet terhadap data citra digital pada daun gambas untuk identifikasi jenis penyakit.
	Hasil Utama	Penerapan Convolutional Neural Network (CNN) dengan arsitektur MobileNet untuk identifikasi penyakit pada tanaman gambas memiliki akurasi 90%.
	Batasan	Hanya 3 jenis penyakit yang diinputkan.
10	Paper ke-10	
	Judul Paper	Insect Identification Among Deep Learning's Meta-architectures Using TensorFlow
	Penulis	Deven J. Patel, Nirav Bhatt
	Jurnal/Konferensi	International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019
	URL	https://www.researchgate.net/publication/337917435_Insect_Identification_Among_Deep_Learning's_Meta-architectures_Using_TensorFlow
	Permasalahan	Pertanian menyediakan makanan bagi keberadaan manusia, dimana serangga merusak tanaman. Identifikasi serangga merupakan proses yang sulit dan tunduk pada pendapat ahli.
	Kontribusi	Menggunakan arsitektur terbaru dan mengukur kinerja akurasi jaringan menggunakan kumpulan data serangga yang kecil. Arsitektur meta CNN, yaitu Faster-RCNN, SSD Inception dan SSD Mobilenet.
	Metode/Solusi	Dalam penelitian ini, meta-arsitektur CNN yang tersebar luas digunakan untuk mengidentifikasi spesies serangga yang berbeda
	Hasil Utama	Eksperimen kami menunjukkan bahwa meta-arsitektur CNN yang canggih Faster-RCNN berkinerja lebih baik dengan akurasi identifikasi setinggi 95,33% dan sensitivitas sebesar 91,06% untuk operasi deteksi hama tertentu.

	Batasan	Dari beberapa jenis serangga, hanya dipilih tiga jenis dalam penelitian itu.
11	Paper ke-11	
	Judul Paper	Klasifikasi Citra Buah Menggunakan Convolutional Neural Network
	Penulis	Febian Fitra Maulana, Naim Rochmawati
	Jurnal/Konferensi	JINACS: Volume 01 Nomor 02, 2019 (Journal of Informatics and Computer Science) ISSN : 2686-2220
	URL	https://jurnalmahasiswa.unesa.ac.id/index.php/jinacs/article/view/31406
	Permasalahan	Mengembangkan arsitektur CNN yang menggunakan objek citra buah-buahan sebagai data uji
	Kontribusi	Model CNN yang menggunakan perpaduan 3 convolutional layer dan 2 hidden layer
	Metode/Solusi	Arsitektur CNN dalam penelitian ini memiliki 3 lapisan konvolusi (convolutional layer) yang menggunakan fungsi aktivasi ReLu dan dipadukan dengan Max Pooling layer. Pada tahap klasifikasi (Fully Connected Layer) digunakan algoritma ANN (Artificial Neural Network) yang memiliki 2 hidden layer.
	Hasil Utama	Akurasi yang didapatkan dari proses testing yang menggunakan 345 citra uji menunjukkan angka 97,97%.
	Batasan	Kelas data yang digunakan yaitu sejumlah 15 kelas dari 111 kelas pada dataset fruit-360.
12	Paper ke-12	
	Judul Paper	Vision-Based Perception and Classification of Mosquitoes Using Support Vector Machine
	Penulis	Masataka Fuchida, Thejus Pathmakumar, Rajesh Elara Mohan, Ning Tan dan Akio Nakamura
	Jurnal/Konferensi	Appl. Sci. 2017, 7, 51; doi:10.3390/app7010051

	URL	https://www.mdpi.com/2076-3417/7/1/51
	Permasalahan	Metode identifikasi nyamuk mencakup analisis DNA, yang membutuhkan metode pengumpulan sampel nyamuk yang sangat belum berkembang dan membutuhkan beberapa pemeriksaan laboratorium dengan pengawasan ahli.
	Kontribusi	Pengembangan sistem klasifikasi nyamuk otomatis, diakhiri dengan hasil eksperimen menggunakan modul sensor prototipe yang memvalidasi pendekatan yang diusulkan dalam mengklasifikasikan nyamuk dari sekumpulan bug lain yang telah ditentukan sebelumnya
	Metode/Solusi	Dalam penelitian ini, kami menggunakan pengklasifikasi mesin vektor pendukung (SVM) untuk mengklasifikasikan spesies nyamuk, lebah, dan lalat. Penelitian ini menggunakan tiga varian pengklasifikasi SVM, dan kinerja varian ini dalam konteks klasifikasi nyamuk dievaluasi. SVM yang digunakan yaitu C-Support Vector Classification (C-SVC) dan varian nu-Support Vector Classification (nu-SVC) dari modul SVM untuk melakukan klasifikasi
	Hasil Utama	Hasil eksperimen yang melibatkan klasifikasi antara nyamuk dan kumpulan bug lain yang telah ditentukan sebelumnya menggunakan strategi klasifikasi ganda menunjukkan kemanjuran dan validitas pendekatan yang diusulkan dengan perolehan maksimum 98%.
	Batasan	Klasifikasi nyamuk hanya 3 jenis.
13	Paper ke-13	
	Judul Paper	Identifikasi Penyakit Pada Daun Tebu dengan <i>Gray Level Cooccurrence Matrix</i> dan <i>Color Moments</i>
	Penulis	Ratih Kartika Dewi, R.V. Hari Ginardi
	Jurnal/Konferensi	Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK) Vol. 1, No. 2, Oktober 2014 hlm. 70-77
	URL	https://jtiik.ub.ac.id/index.php/jtiik/article/view/114
	Permasalahan	Karat dan mosaik adalah penyakit yang menyerang

		tebu di Indonesia dan menimbulkan kerugian. Penyakit karat pernah menyebabkan penurunan panen sebanyak 40% di Florida pada 1988 (Raid, 2006).
	Kontribusi	Fitur tekstur dengan gray level co-occurrence matrix dan fitur warna dengan color moments yang dipadukan dengan metode klasifikasi SVM diharapkan cocok untuk identifikasi penyakit pada daun tebu.
	Metode/Solusi	Klasifikasi daun tebu berpenyakit dilakukan berdasarkan fitur yang telah diekstraksi pada tahap sebelumnya. Penelitian ini menggunakan metode support vector machine (SVM) karena SVM cocok untuk klasifikasi yang datanya sedikit (Li, 2010).
	Hasil Utama	Kombinasi fitur tekstur dengan GLCM correlation, energy, homogeneity dan variance bersama fitur warna dengan color moments 1,2 dan 3 yang diuji pada skenario 4 merupakan kombinasi fitur yang direkomendasikan untuk identifikasi penyakit pada daun tebu dengan akurasi 97%.
	Batasan	Identifikasi citra digital daun dengan pemilihan fitur tekstur dan warna melalui gray level co-occurrence matrix (GLCM) dan color moments
14	Paper ke-14	
	Judul Paper	Implementasi <i>Convolutional Neural Networks</i> (CNN) untuk Klasifikasi Ekspresi Citra Wajah pada Fer-2013 Dataset
	Penulis	Derry Alamsyah, Dicky Pratama
	Jurnal/Konferensi	Jurnal Teknologi Informasi Vol.4, No.2, Desember 2020
	URL	http://jurnal.una.ac.id/index.php/jurti/article/view/1714
	Permasalahan	Pengenalan ekspresi wajah menjadi penting seiring dengan meningkatnya teknologi yang ada pada komputer, telepon genggam, robot dan sebagainya.
	Kontribusi	Implementasi sistem ini menggunakan 4 jenis optimizer yaitu Adam, Adamax, N-adam, dan SGD.

	Metode/Solusi	Rancang sistem yang dibuat dengan arsitektur CNN yang terdiri dari tiga convolutional layer dengan masing-masing fungsi aktivasi ReLu dan max pooling pada pooling layer. Sementara itu terdapat 2 dense layer dengan fungsi aktivasi ReLu dan sigmoid.
	Hasil Utama	Kemampuan CNN dalam mengenali emosi melalui citra wajah paling baik menggunakan Adamax optimizer dengan nilai akurasi sebesar 66%.
	Batasan	Implementasi sistem ini menggunakan 4 jenis optimizer yaitu Adam, Adamax, N-adam, dan SGD. Sementara itu data dibagi kedalam data latih dan validasi dalam proses pelatihannya dengan menggunakan 10 epoch.
15	Paper ke-15	
	Judul Paper	Implementasi Deep Learning untuk Image Classification Menggunakan Algoritma <i>Convolutional Neural Network</i> (CNN) pada Citra Kebun Dan Sawah
	Penulis	Meiga Isyatan Mardiyah
	Jurnal/Konferensi	(Skripsi)
	URL	https://dspace.uii.ac.id/handle/123456789/28083
	Permasalahan	Sawah yang merupakan contoh pertanian dan perkebunan yang merupakan contoh kebun memiliki karakteristik yang sama sehingga sulit untuk dibedakan oleh masyarakat atau orang awam apabila dilihat dengan menggunakan foto udara atau foto yang diambil dari ketinggian tertentu.
	Kontribusi	Penelitian ini menguji tiga perbandingan data dan dipilih satu yang terbaik untuk dijadikan testing.
	Metode/Solusi	Peneliti memanfaatkan metode CNN ini untuk dapat mengklasifikasi kebun dan sawah karena memiliki lapisan konvolusi yang memiliki fungsi pembelajaran citra menjadi lebih efisien untuk diimplementasikan.
	Hasil Utama	Skenario perbandingan dataset train sebesar 80% dan dataset test sebesar 20%. Hasil akurasi yang diperoleh yakni sebesar 0.9625 atau 96.25% untuk akurasi training dan sebesar 0.75 atau 75% untuk akurasi

		testing. Hasil dari akurasi pengujian yang didapatkan dengan 20 data uji berdasarkan arsitektur terbaik untuk pengklasifikasian citra kebun dan citra sawah sesuai kategorinya sebesar 75%.
	Batasan	Citra yang digunakan memiliki ukuran 32x32 piksel, Sampel yang digunakan dalam penelitian ini sebanyak 100 citra sawah dan citra kebun, setiap kategori (sawah dan kebun) terdiri atas 50 citra.
16	Paper ke-16	
	Judul Paper	Klasifikasi Motif Batik Menggunakan Convolutional Neural Network
	Penulis	Rizki Mawan
	Jurnal/Konferensi	Jurnal Open Access Yayasan Lentera Dua Indonesia
	URL	https://lenteradua.net/jurnal/index.php/jnanaloka/article/view/2
	Permasalahan	Banyaknya pola batik di Indonesia mengakibatkan sulitnya untuk mengidentifikasi motif, khususnya bagi masyarakat awam. Adanya suatu metode untuk mempermudah pengenalan pola batik tentu memiliki banyak manfaat.
	Kontribusi	Pada penelitian ini <i>grayscale</i> digunakan untuk proses perubahan warna motif batik.
	Metode/Solusi	Citra pada batik memiliki beberapa layer sehingga penggunaan metode <i>Convolutional Neural Network</i> (CNN) sangat baik mengingat pada CNN memiliki <i>Multi Layer Perceptron</i> (MLP). Dalam tulisan ini, digunakan model klasifikasi batik berdasarkan metode CNN dan CNN kombinasi <i>Gray Scale</i> .
	Hasil Utama	Akurasi yang didapatkan menggunakan metode CNN dan CNN kombinasi dengan <i>Grayscale</i> memiliki tingkat akurasi yang berbeda. Nilai akurasi terbaik didapatkan dengan metode CNN kombinasi <i>grayscale</i> dengan tingkat akurasi sebesar 70%.
	Batasan	Jumlah kelas pada penelitian ini yaitu 3 kelas

17	Paper ke-16	
Judul Paper	Implementasi Metode CNN dalam Klasifikasi Gambar Jamur pada Analisis Image Processing (Studi Kasus: Gambar Jamur dengan Genus Agaricus dan Amanita)	
Penulis	Ocktavia Nurima Putri	
Jurnal/Konferensi	(Skripsi)	
URL	https://dspace.uui.ac.id/handle/123456789/23677?show=full	
Permasalahan	Di Eropa Utara, terdapat beberapa genus jamur yang banyak tumbuh dan berkembang dengan luas, diantaranya adalah Agaricus yang dikenal sebagai jamur layak konsumsi seperti Agaricus Bisporus (jamur kancing), dan genus jamur Amanita, spesies dari genus jamur tersebut dikenal sebagai jamur paling beracun di dunia, salah satunya disebut sebagai death caps yang dapat menyebabkan iritasi dan berpotensi mematikan. Namun morfologi dari beberapa jamur beracun ini bisa menyerupai jamur yang dapat dikonsumsi	
Kontribusi	Percobaan menggunakan 3 jenis optimizer, yaitu Adam, RMSProp, dan SGD.	
Metode/Solusi	Kemampuan CNN diklaim sebagai model terbaik untuk memecahkan permasalahan object detection dan object recognition. Maka penelitian ini menggunakan metode CNN untuk klasifikasi gambar jamur berdasarkan jenis genus.	
Hasil Utama	Diperoleh perbandingan akurasi dari 3 jenis optimizer yaitu 62% untuk optimizer Adam dengan epoch 100, 54% untuk optimizer RMSProp dengan epoch 50, dan 55% untuk optimizer SGD dengan epoch 50.	
Batasan	Data yang digunakan merupakan data gambar jamur dengan 2 macam genus dengan total data yang digunakan adalah sebanyak 520 gambar jamur.	