

# Daftar Isi

[Daftar Isi 2](#_Toc130171562)

[Bagian I 3](#_Toc130171563)

[Introduction 3](#_Toc130171564)

[Pengertian Pengolahan Citra 4](#_Toc130171565)

[Sistem Pencitraan 4](#_Toc130171566)

[Objek dan Citra 4](#_Toc130171567)

[Pengolahan Citra Digital 4](#_Toc130171568)

[Aplikasi Pengolahan Citra 4](#_Toc130171569)

[Bagian II 5](#_Toc130171570)

[Konsep Pengolahan Citra 5](#_Toc130171571)

[Persepsi Citra pada Manusia dan Komputer 5](#_Toc130171572)

[Citra Film sinar-X 6](#_Toc130171573)

[Citra 2D dan 3D 6](#_Toc130171574)

[Digitalisasi Citra 6](#_Toc130171575)

[Kualitas Citra 6](#_Toc130171576)

[Pengolahan Citra berbasis Artificial Intelligence 6](#_Toc130171577)

[Bagian III 9](#_Toc130171578)

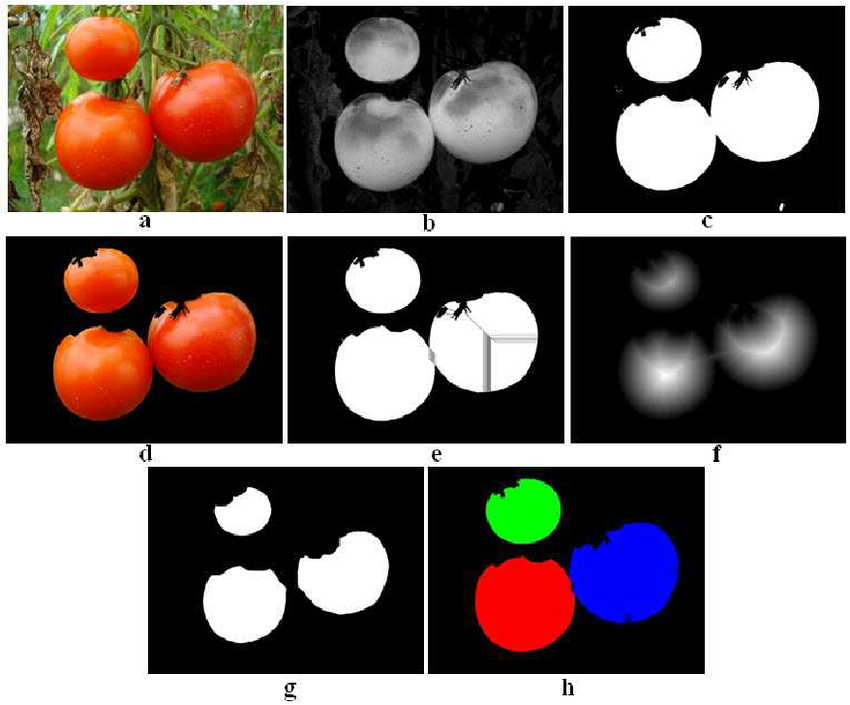
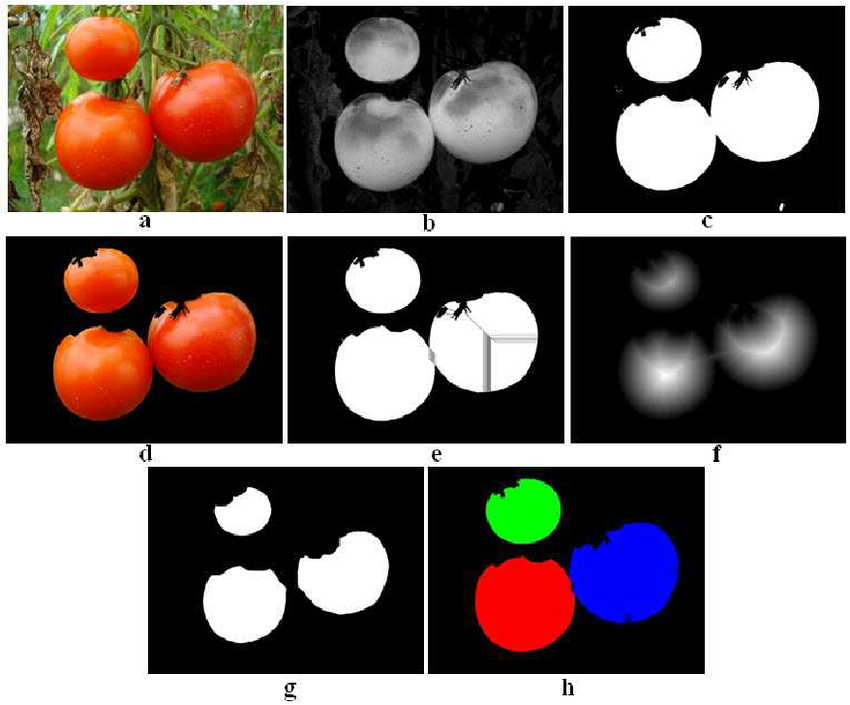
[Image Classification 9](#_Toc130171579)

[Bagian IV 16](#_Toc130171580)

[Image Segmentation 16](#_Toc130171581)

## Bagian I

## Introduction



“Dalam mempelajari pengolahan citra, tentu kita membutuhkan pemahaman dasar yang baik. Dalam bagian pertama akan dijelaskan terkait dasar-dasar pengolahan citra. Mulai dari pengertian dan konsep awal pengolahan citra, pengembangan metode pengolahan citra dengan deep learning, hingga berbagai macam hal yang dapat dilakukan dengan pengolahan citra.”

### Pengertian Pengolahan Citra

Sejarah dan pengertian pengolahan citra

### Sistem Pencitraan

Membahas system kerja pencitraan, konsep gelombang pada rentang visible light

### Objek dan Citra

Membahas proses akuisisi data citra dari objek

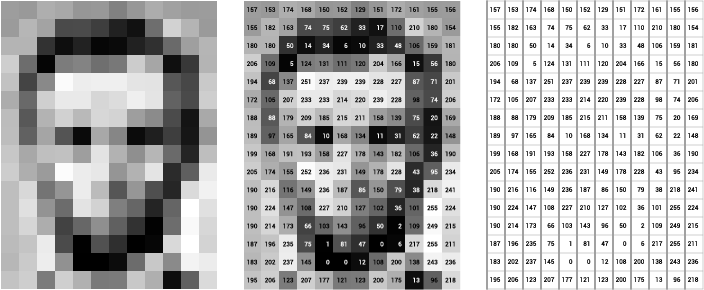
### Pengolahan Citra Digital

Proses konversi sinyal analog to digital, jenis pengolahan citra digital (extraction, enhancement, synthesis, restoration, translation)

### Aplikasi Pengolahan Citra

## Bagian II

## Konsep Pengolahan Citra

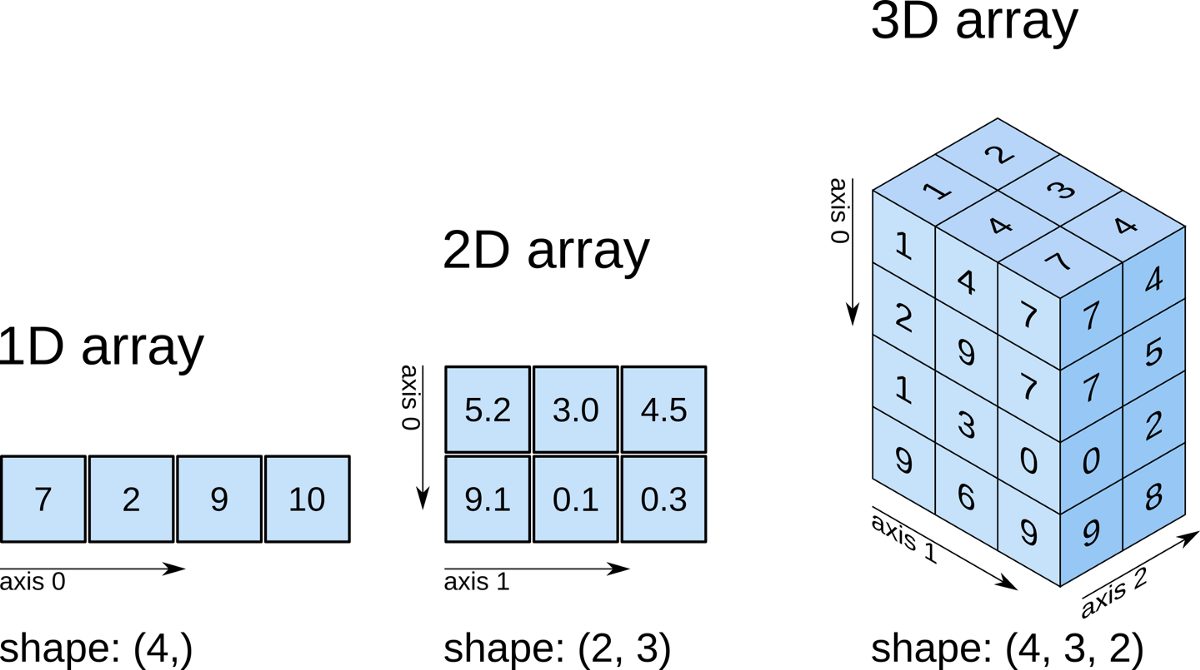


“Pengolahan citra telah mengalami perkembangan yang pesat seiring dengan aplikasinya yang sangat beragam dalam berbagai bidang. Untuk memahami pengolahan citra dengan baik, sangat penting untuk memahami konsep dasar pengolahan citra. Konsep dasar pengolahan citra mencakup pemahaman tentang struktur dasar citra digital hingga Teknik pengolahan citra yang lebih kompleks seperti segmentasi citra dan klasifikasi citra. Dalam bab ini akan dipelajari konsep dasar dalam pengolahan citra.”

### Persepsi Citra pada Manusia dan Komputer

### Citra Film sinar-X

### Citra 2D dan 3D



### Digitalisasi Citra

### Kualitas Citra

### Pengolahan Citra berbasis Artificial Intelligence

### Edge Detection

Graphical user interface

Description automatically generated

Gambar. Visualisasi edge detection pada citra.

Metode pengolahan citra telah berkembang sangat pesar di dekade ini, hal ini tidak lepas dari berkembangnya perangkat komputer yang semakin canggih dan terjangkau. Manfaat pengolahan citra sangat banyak di berbagai bidang, termasuk untuk bidang falak khususnya rukyat (pengamatan). Pengolahan citra sangat penting dilakukan untuk mengolah citra pengamatan maupun mendapatkan informasi dari citra yang didapatkan. Artikel ini adalah artikel pembuka untuk seri Image Processing for Falakiyah. Dalam artikel ini akan dibahas penerapan sederhana metode Deteksi Tepi atau Edge Detection untuk citra dengan Python.

Ada 3 tools utama yang akan digunakan. Yaitu bahasa pemrograman Python, Library OpenCV, dan Library matplotlib. OpenCV adalah Library yang dibuat khusus untuk mengolah citra dengan beragam metode sesuai kebutuhannya, nantinya kita akan memanfaatkan OpenCV untuk memanggil fungsi Edge Detection. Library matplotlib digunakan untuk membuat grafik ataupun menampilkan citra yang telah diolah. Deteksi tepi dalam beberapa penelitian digunakan untuk mendeteksi berbagai fenomena alam, termasuk dalam upaya deteksi tepi pada citra medis. Tujuan dilakukan filtering citra medis ini diantaranya untuk mempertajam citra medis yang dihasilkan oleh pesawat sinar-X.



Gambar. Contoh hasil deteksi tepi pada citra sinar-X tulang.

Terlihat bahwa penerapan filter Canny memudahkan kita untuk mendeteksi seberapa tajam garis batas pada citra sinar-X. Pada aplikasi pemrosesan citra sinar-X, penting untuk dapat mengekstrak informasi yang relevan dari citra tersebut. Salah satu informasi yang dapat diekstrak adalah tingkat ketajaman atau kejelasan dari batas-batas yang terdapat pada citra sinar-X tersebut. Dalam hal ini, filter Canny dapat menjadi salah satu alat yang sangat berguna untuk mempermudah proses deteksi batas pada citra sinar-X. Dengan menggunakan filter Canny, citra sinar-X akan diproses dan dihasilkan citra baru yang hanya menunjukkan ketajaman batas pada citra asli.

Citra hasil pemrosesan menggunakan filter Canny ini kemudian dapat digunakan untuk analisis lebih lanjut, seperti untuk menghitung panjang atau sudut dari batas-batas yang terdeteksi, atau untuk mengidentifikasi struktur atau objek yang terdapat dalam citra. Dalam keseluruhan proses pemrosesan citra sinar-X, filter Canny memainkan peran penting dalam membantu mempermudah dan meningkatkan efisiensi dalam mendeteksi dan mengekstrak informasi dari citra sinar-X tersebut.

#1. Import Library

import cv2

import matplotlib.pyplot as plt

#2. Input Citra yang Akan Diproses

im = cv2.imread('/content/image\_1.jfif') #Sesuaikan dengan nama file

im = cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)

plt.imshow(im, cmap='gray')

plt.show()

#3. Memanggil dan Menerapkan Edge Detection

#Menerapkan filter canny

filter\_canny = cv2.Canny(im,25,255,L2gradient=False)

#Menerapkan filter sobel

filter\_sobel = cv2.Sobel(src=im, ddepth=cv2.CV\_8U, dx=1, dy=1, ksize=5)

#4. Plot Hasil Deteksi Tepi

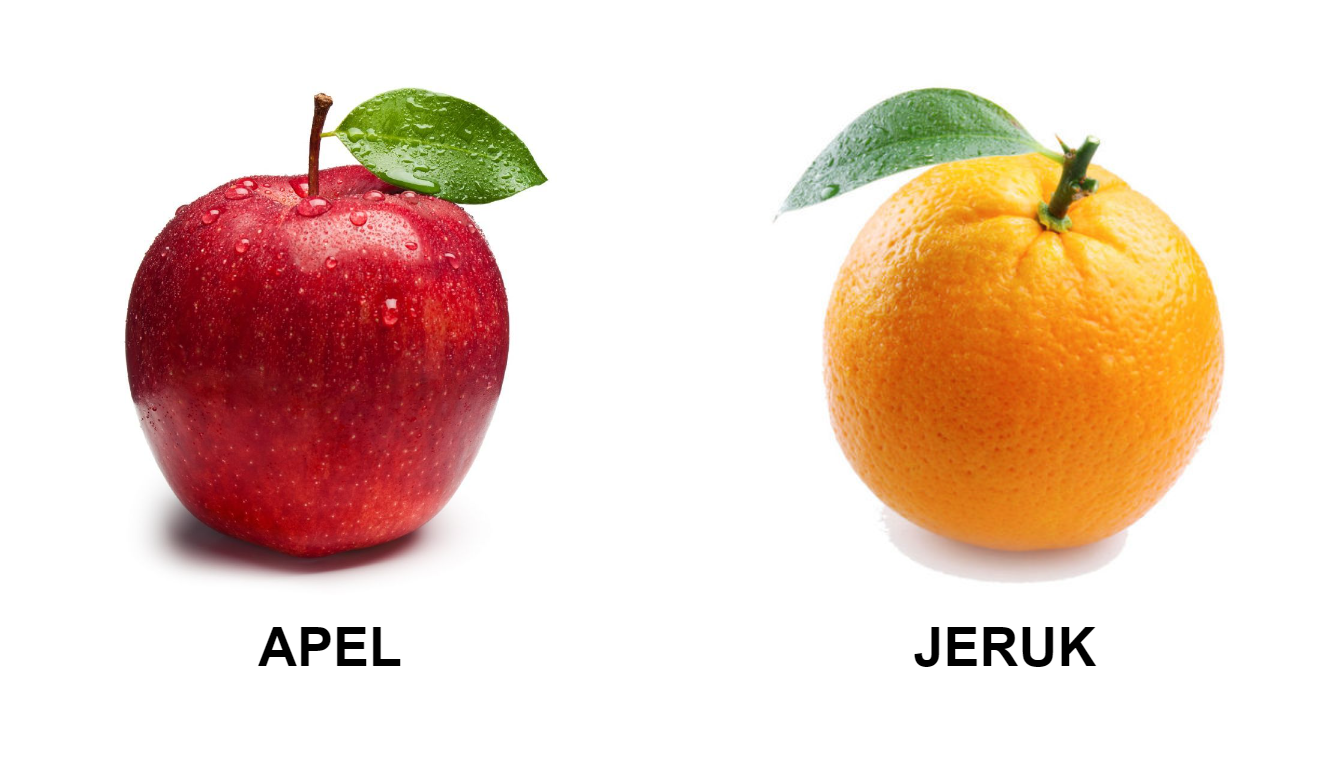
plt.imshow(filter\_canny, cmap='gray')

plt.show()

Dalam pemilihan filter edge detection, ada beberapa parameter yang perlu diperhatikan dan disesuaikan. Hal ini akan menentukan bagaimana kualitas hasil deteksi tepi citra yang didapatkan. Full code untuk program ini dapat diakses pada google colaboratory research berikut : <https://colab.research.google.com/drive/1-muHa46OtiyJRX-sTm8p1WH6vnb8N82u?usp=sharing>

## Bagian III

## Image Classification



“Penggunaan metode klasifikasi citra dapat memungkinkan mengidentifikasi objek atau pola pada suatu citra dengan akurasi tinggi secara efisien. Hal ini dapat membantu kita dalam pengambilan keputusan yang lebih baik serta meningkatkan efisiensi dalam pengolahan citra. Oleh karena itu, penting untuk memahami konsep dasar metode klasifikasi citra dan juga bagaimana metode ini diterapkan dalam bidang medis.”

### Pendahuluan

Klasifikasi citra adalah suatu proses pengolahan citra untuk menentukan klasifikasi citra berdasarkan kelas-kelas tertentu. Klasifikasi citra merupakan salah satu bagian dari bidang *Komputer Vision* yang memiliki manfaat besar dan banyak diaplikasikan dalam kegiatan sehari-hari, terutama di era Revolusi Industri 4.0 sekarang.[1]

Proses klasifikasi citra dapat dilakukan dengan berbagai macam metode, mulai dari metode klasifikasi sederhana berdasarkan karakteristik citra hingga metode yang lebih maju seperti pengolahan citra digital berbasis *Machine Learning*. Setiap citra yang akan diolah dengan *Machine Learning* akan dibagi menjadi *training dataset, validation dataset*, dan  *testing dataset*. Kemudian, setiap citra juga diberi label atau penamaan sesuai citra. Saat proses *training*, label tersebut akan menjadi perbandingan dengan hipotesis yang diberikan oleh model *Machine Learning* dan akan menampilkan nilai *error*. Klasifikasi citra dengan metode seperti ini akan sangat efektif dan akurat karena dapat diketahui nilai kesalahan pada model sekaligus dapat memperbaiki model di saat bersamaan hingga didapatkan model yang ideal.

An orange and an apple

Description automatically generated with medium confidence

Gambar. Klasifikasi citra Apel dan Jeruk.

### Convolutional neural network

*Convolutional neural network* (CNN) atau bisa juga disebut sebagai Jaringan Syaraf Konvolusi telah ada sejak 1980-an dimana dikembangkan sebuah algoritma untuk mengenali suatu *Input* berupa citra. Pada saat itu, CNN pertama yang dibuat berhasil menganilasi citra *Input* 2 Dimensi yang berisi angka-angka. Kemudian CNN terus berkembang pesat sejak 2012-2013 ketika revolusi industry mulai ramai berkembang [2].

Diagram

Description automatically generated

Gambar. Arsitektur dalam *Convolutional neural network*

Sebagaimana ditunjukkan pada Gambar 2.8., arsitektur CNN terdiri dari beberapa lapisan atau *layer*, yang meliputi *Activation, Batch Normalization, Drop Out, Convolution Layer, Pooling layer,* dan *Fully Connected.* Saat ini ada banyak arsitektur CNN yang dibuat dan dibangun dengan dasar dari *Convolution Layer, Pooling layer,* dan *Fully Connected*. Setiap arsitektur CNN yang dibangun disesuaikan dengan bentuk data masukan dan hasil keluaran yang diharapkan. Oleh karena itu di setiap arsitektur CNN digunakan susunan *layer* yang berbeda-beda dengan parameter yang berbeda-beda juga. Hal ini semua tergantung dari perancang arsitektur bagaimana membangun arsitektur CNN. Pada umumnya untuk skala penelitian akademis, arsitektur CNN yang dibangun bertujuan untuk mendapatkan nilai akurasi, presisi, recall, dan f1 score yang paling tinggi. Sedangkan untuk skala industri, arsitektur CNN yang dibangun berfokus pada performa yang tinggi dengan beban komputasi yang paling minimum. Hal ini dilakukan untuk mengrangi biaya dalam aplikasi CNN di dunia industri. Karena semakin tinggi beban komputasi, maka kebutuhan perangkat keras pemroses dan waktu pemrosesan juga akan semakin tinggi. Hal ini akan berdampak pada meningkatnya biaya operasional [2].

### *Convolution Layer*

Shape

Description automatically generated

Gambar. Ilustrasi proses konvolusi citra.

Dalam bidang *Komputer Vision*, istilah konvolusi yang digunakan sedikit berbeda dengan teknik konvolusi matriks dalam ilmu Matematika. Operasi konvolusi yang dimaksud dan dilakukan dalam bidang *Computer Vision* sebenarnya adalah operasi *Cross-Correlation* sebagaimana diilustrasikan pada Gambar 2.9. Ada sedikit perbedaan diantara kedua operasi tersebut. Pada *Cross-Correlation* tidak dilakukan proses perputaran pada matriks, namun hal ini dilakukan dengan melakukan perkalian antara kotak di kedua matriks secara langsung dan menjumlahkannya. Namun pada bidang *Komputer Vision* hal ini disebut konvolusi, karena menurut Prof. Andrew N. G. hal tersebut tidak berpengaruh secara signifikan pada proses pengolahan citra dalam CNN. Sehingga, metode konvolusi citra dapat dituliskan dengan persamaan :

……..(2.2.)

Secara sederhana visualisasi proses konvolusi antara 2 matriks (matriks *Input* dan matriks filter/kernel) ditunjukkan pada Gambar.

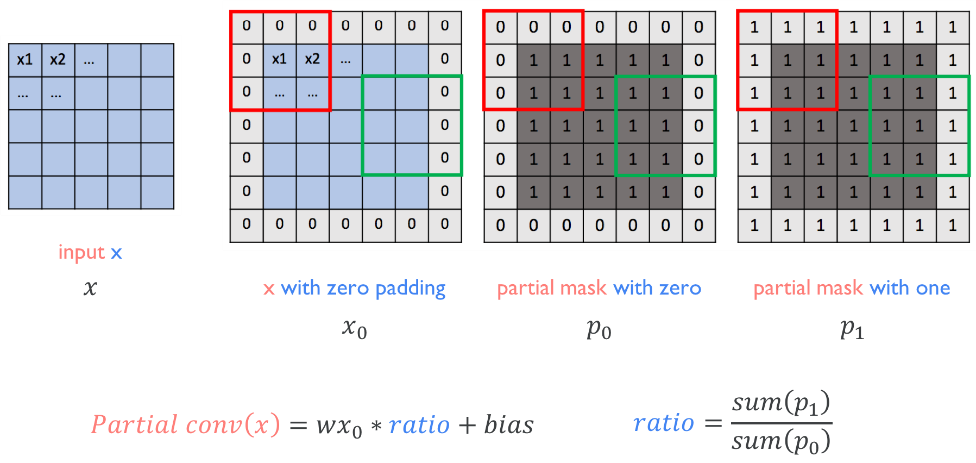
Table

Description automatically generated with medium confidence

Gambar. Proses operasi Konvolusi citra

Ukuran dan variable di dalam filter atau kernel termasuk dalam parameter yang menentukan hasil konvolusi nantinya. Filter ini dapat berguna sebagai pendeteksi garis vertikal, pendeteksi garis horizontal, pendeteksi bentuk kotak, dan banyak lagi. Hal ini menjadi salah satu penentu pada hasil akhir arsitektur CNN yang dibangun [3].

Operasi konvolusi seperti ini akan mengurangi ukuran *layer* sesuai dengan ukuran *filter* yang digunakan. Pada operasi konvolusi, ada istilah *padding Convolution* dan *strided Convolution*. *Padding Convolution* adalah proses penambahan ukuran *layer* dengan matriks yang berisi variable tertentu. Penambahan ini bertujuan agar ukuran *layer* tidak berkurang setelah melalui proses konvolusi. Ilustrasi dari proses *padding Convolution* ditunjukkan oleh Gambar.



Gambar. *Padding Convolution.*

*Strided Convolution* adalah proses memberikan jeda pada pergeseran matriks ketika proses konvolusi. Penggunaan *Strided Convolution* berdampak pada penurunan ukuran *layer* yang lebih banyak sehingga mengurangi beban komputasi pada proses perhitungan dalam algoritma *Convolution Neural network* [3].

Table

Description automatically generated with medium confidence

Gambar. Ilustrasi proses konvolusi citra dengan *Stride* =2*.*

### *Pooling layer*

*Pooling layer* adalah salahsatu metode untuk mengurangi ukuran *layer*. *Pooling layer* bekerja dengan cara mengambil parameter yang menonjol sebagai interpretasi dari citra asli. Sebagaimana ditunjukkan pada Gambar, ada dua macam yang umumnya digunakan, yaitu *Max-Pool Layer* dan *Average-Pooling layer*. Seperti namanya, *Max-Pool Layer* bekerja dengan mengambil parameter maksimum di suatu area matriks. Sedangkan *Average-Pooling layer* bekerja dengan cara yang sama namun yang diambil adalah nilai rata-rata pada parameter yang ada [3].

A picture containing table

Description automatically generated

Gambar. *Max-Pooling* dan *Average-Pooling*.

### *Fully connected layer*

*Diagram

Description automatically generated*

Gambar. *Fully connected layer.*

*Fully connected layer* dipakai pada *layer* terakhir dalam suatu arsitektur CNN. *Fully connected layer* berbentuk matriks 1 dimensi, seperti serangkaian angka yang berjejer panjang (lihat Gambar 2.14). *Fully connected layer* bekerja seperti *neuron* pada umumnya dan memiliki hidden layer, Activation Function, Output layer, dan loss function. Berfungsi sebagai gerbang Output pada arsitektur CNN yang telah dibuat [3].

### 

### Arsitektur UBNet

Arsitektur UBNet adalah arsitektur CNN yang dikembangkan dengan tujuan awal focus untuk mengolah citra sinar-X dada dalam bentuk citra 2D. UBNet dikembangkan dengan tujuan mendapatkan arsitektur CNN yang ringan dan cepat dalam melakukan klasifikasi citra sinar-X. UBNet terdiri dari UBNet v1, UBNet v2, dan UBNet v3. UBNet v1 terdiri dari 7 layer konvolusi dan 3 layer artificial neural network. UBNet v2 terdiri dari 4 layer konvolusi dan 3 layer artificial neural network. Sedangkan UBNet v3 adalah gabungan dari UBNet v1 dan UBNet v2 yang berfungsi untuk klasifikasi Covid-19. Visualisasi UBNet v1 ditampilkan pada Gambar 2.18. Input citra pada UBNet v1 standar berukuran 400x400x1 [4].

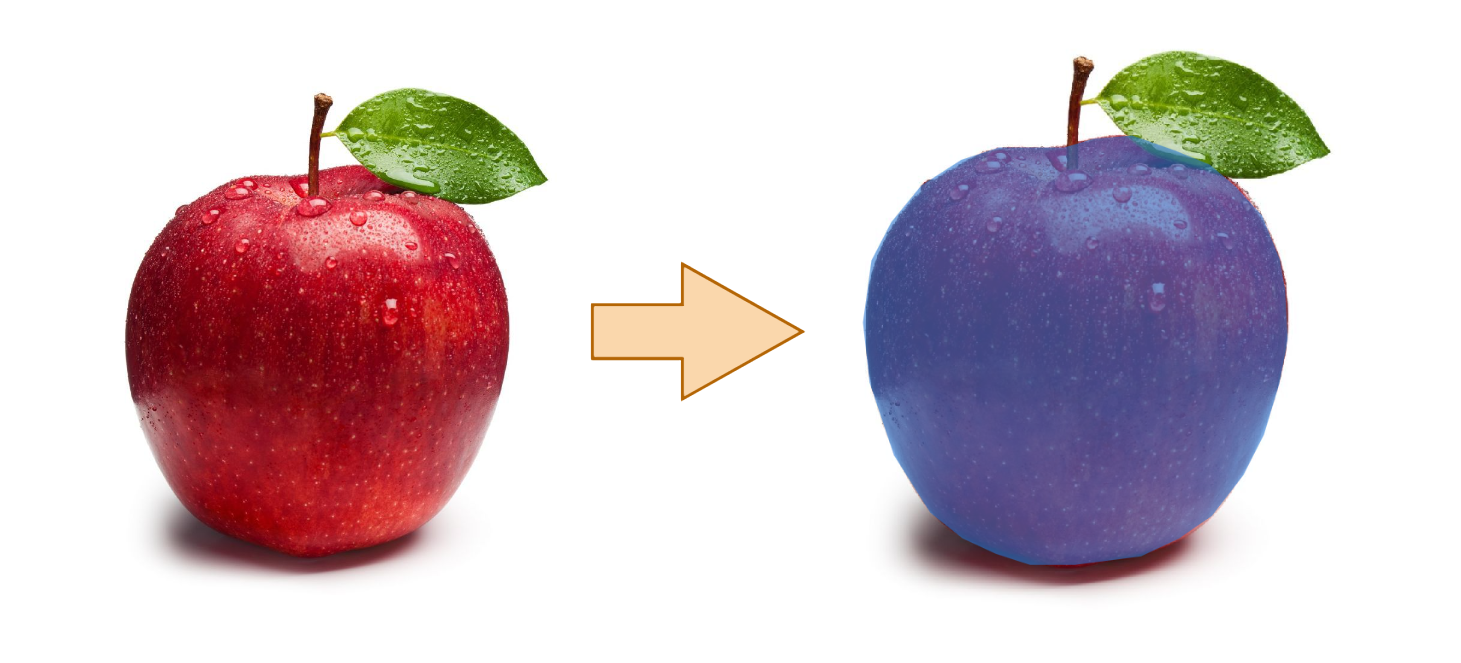
A picture containing diagram

Description automatically generated

Gambar. Arsitektur UBNet v1.

## Bagian IV

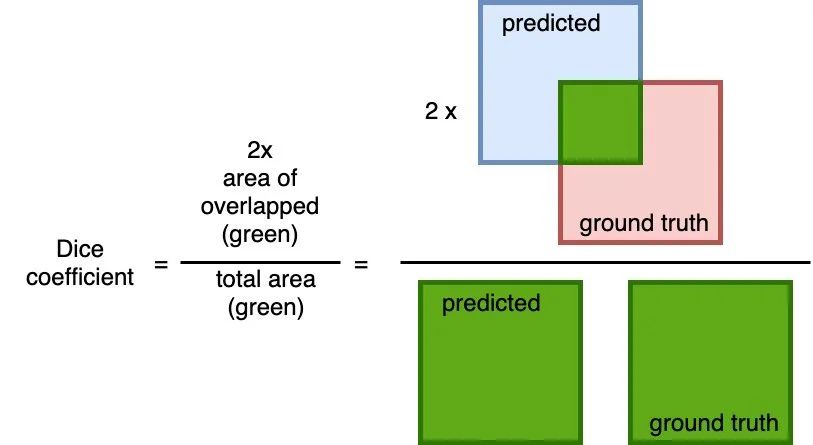
## Image Segmentation



“Segmentasi citra adalah metode penting dalam pengolahan citra yang memungkinkan kita untuk memisahkan dan mengidentifikasi area suatu objek dalam citra dengan akurasi tinggi. Teknik segmentasi citra memungkinkan kita memperoleh informasi yang lebih spesifik dan detail terkait suatu citra. Oleh karena itu, memahami metode segmentasi citra dan penerapannya sangatlah penting dalam bidang medis.”

### Metode Evaluasi : Dice Coefficient (DSC)

Setelah model telah jadi, maka perlu dilakukan pengujian agar dapat diketahui seberapa baik performa model tersebut. Ada beberapa metode dalam evaluasi performa model, salahsatunya adalah *Dice Coefficient*. Prinsip dasar dari *Dice Coefficient* adalah mengukur seberapa dekat hasil prediksi segmentasi citra jika dibandingkan dengan *ground truth* dari segmentasi citra input. *Dice Coefficient* banyak digunakan sebagai metode evaluasi untuk metode segmentasi citra karena cukup sederhana dalam perumusannya dan juga dapat menggambarkan dengan baik performa model [5].



Gambar. Prinsip dasar dan perhitungan *Dice Coefficient.*

# Daftar Pustaka

[1] F. Luo, Y. Huang, W. Tu, and J. Liu, “Local manifold sparse model for image classification,” *Neurocomputing*, no. xxxx, 2019, doi: 10.1016/j.neucom.2019.11.084.

[2] A. Geron, *Hands-on Machine Learning with Sikit-Learn, Keras & TensorFlow*. California: O"Reilly Media, Inc., 2019.

[3] A. Rosebrock, *Deep Learning for Computer Vision with Python*, no. c. USA: PYIMAGESEARCH, 2017.

[4] C. S. Widodo, N. Agus, M. M. Mahasin, Y. Yueniwati, T. A. Putranto, and P. I. Patra, “UBNet: Deep learning-based approach for automatic X-ray image detection of pneumonia and COVID-19 patients,” *J. Xray. Sci. Technol.*, 2021, doi: 10.3233/XST-211005.

[5] K. H. Zou *et al.*, “Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index,” *Acad. Radiol.*, vol. 11, no. 2, pp. 178–189, 2004, doi: 10.1016/S1076-6332(03)00671-8.