

BAB II

STUDI PUSTAKA

2.1 Tinjauan Pustaka

Penelitian tentang klasifikasi buah dengan menggunakan pengolahan citra sudah banyak dilakukan. Penelitian-penelitian tersebut menggunakan berbagai macam metode salah satunya dalam penelitian Suchitra A. Khoje, dkk., (2013) tentang “*Automated Skin Defect Identification System for Fruit Grading Based on Discrete Curvelet Transform*” bertujuan mengembangkan metodologi untuk menilai kualitas permukaan buah secara obyektif dengan menggunakan analisis tekstur berdasarkan transformasi *curvelet* dengan resolusi citra rendah dan tinggi. Pengukuran tekstur berdasarkan transformasi *curvelet* menggunakan 4 ekstraksi ciri diantaranya *energy*, *entropy*, rata-rata (*mean*) dan standar deviasi (*std*). Keempat ekstraksi ciri tersebut menjadi variabel yang digunakan untuk menganalisa kualitas permukaan buah. Metode yang digunakan untuk mengklasifikasikan kualitas permukaan buah yaitu klasifikasi *Support Vector Machine* (SVM) dan *Probabilistik Neural Networks* (PNN) dengan membagi dua kategori yaitu sehat dan cacat. Hasil penelitian menunjukkan bahwa akurasi terbaik dari klasifikasi SVM diperoleh dengan akurasi nilai 96%.

Ranjit K N. dkk., (2016) dalam penelitiannya “*Identification and Classification of Fruit Diseases*” penelitian ini bertujuan untuk mengidentifikasi dan mengklasifikasi penyakit buah. Dalam penelitian ini citra buah tersegmentasi menggunakan algoritma *K-means* dan *C-meansclustering*. Kemudian evaluasi kinerja algoritma segmentasi dilakukan dengan menggunakan parameter seperti

Measure of over segmentation (MOS), Measure of under-segmentation (MUS), Measure of overlapping (MOL), Dice similarity measure (DSM), Error-rate (ER). Hasil klasifikasi dari penelitian ini memiliki akurasi yang relatif lebih tinggi dalam semua kasus bila tersegmentasi menggunakan *K-means* dibandingkan algoritma *C-meansclustering*.

Algoritma pembelajaran yang diimplementasikan dalam penelitian ini yaitu menggunakan salah satu arsitektur *deep learning* yaitu *Convolutional Neural Network (CNN)* yang digunakan untuk memecahkan masalah klasifikasi data karena memiliki tingkat akurasi yang tinggi. Penerapan metode CNN pernah dilakukan dalam penelitian R. Socher, dkk (2012). Di mana dalam penelitiannya "*Convolutional-Recursive Deep Learning for 3D Object Classification*" mengkombinasikan jaringan syaraf *convolutional* dan rekursif (CNN dan RNN) sebagai fitur pembelajaran dan klasifikasi gambar RGB-D. Lapisan CNN mempelajari fitur invarian bawaan tingkat rendah yang kemudian sebagai *input* dari beberapa *fixed-tree* RNN yang digabungkan menjadi fitur yang tinggi. Dari penelitian tersebut menghasilkan RNN dengan bobot yang acak sebagai fitur yang canggih sehingga diperoleh kinerja yang mutakhir pada *dataset* objek standar RGB-D menjadi lebih akurat dan cepat selama *training* dan *testing* dibandingkan arsitektur yang sebanding seperti CNN dengan dua lapisan.

I Wayan Suartika E. P., dkk (2016) dalam penelitiannya "Klasifikasi Citra Menggunakan *Convolutional Neural Networks (CNN)* pada Caltech 101" menerapkan salah satu metode *deep learning* yaitu dengan CNN untuk mengklasifikasi citra objek. Dalam metode penelitian yang digunakan dalam algoritma CNN terdapat 3 lapisan yang diimplementasikan, yaitu *convolutional*

layer, *subsampling layer*, dan *fully connected layer*. Metode dalam penelitiannya terlebih dahulu dilakukan proses pengolahan pada data citra untuk memfokuskan objek yang akan diklasifikasi dengan metode *wrapping* dan *cropping*. Selanjutnya melakukan proses *training* dan *testing* dengan menggunakan algoritma CNN. Hasil uji coba dari klasifikasi citra objek dengan tingkat *confusion* yang berbeda pada basis data Caltech 101 menghasilkan nilai akurasi sebesar 20% hingga 50%.

M. Zufar (2016) dalam penelitiannya “*Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time*” menggunakan salah satu metode *deep learning* yaitu *Convolutional Neural Networks* (CNN). Dalam penelitiannya metode tersebut diimplementasikan dengan bantuan *library* OpenCV untuk mendeteksi multi wajah dan perangkat Web Cam M-Tech 5MP. Dalam penelitiannya uji coba yang dilakukan menggunakan konstruksi model CNN sampai kedalaman tujuh lapisan. Klasifikasi dilakukan dengan data *input* dari hasil ekstraksi *Extended Local Binary Pattern* dengan radius 1 dan neighbor 15. Dengan mengimplementasi metode ini, hasil penelitian menunjukkan kinerja pengenalan wajah meraih rata-rata tingkat akurasi lebih dari 89% dalam kurang lebih 2 *frame* per detik.

Beberapa tinjauan pustaka diatas merupakan referensi untuk mengembangkan penelitian ini. Dari hasil analisa berbagai penelitian yang sudah dilakukan terdahulu akan dilakukan riset tentang penerapan pengolahan citra untuk mendeteksi kecacatan permukaan buah manggis menggunakan metode *Deep Learning*. Metode *deep learning* ini digunakan untuk memecahkan masalah klasifikasi data karena penggunaannya yang memiliki proses komputasi yang relatif cepat, sehingga penelitian ini dapat menghasilkan deteksi cacat pada

permukaan buah manggis berbasis pengolahan citra ini dapat membantuk dalam meningkatkan kualitas buah manggis.

2.2 Landasan Teori

2.2.1 Buah Manggis

Manggis memiliki nama latin *Garcinia Mangostana L.* adalah tanaman yang tumbuh di daerah tropis dengan suhu hangat dan stabil. Buahnya berwarna hijau pucat pertama kali lalu berwarna merah keunguan ketika matang. Buah yang dikenal sebagai ratu buah dalam perdagangan ini mengandung aktivitas antiinflamasi dan antioksidan. Sehingga di luar negeri buah manggis ini menjadi buah dengan ekspor terbesar di Indonesia yang dikenal sebagai buah dengan memiliki kadar antioksidan tertinggi di dunia (Wikipedia, 2016).

2.2.2 Pengertian Citra Digital

Citra digital adalah sebuah representasi numerik (biasanya biner) dari gambar dua dimensi. Sebuah gambar dapat didefinisikan sebagai fungsi dua dimensi $f(x, y)$, di mana x dan y adalah koordinat bidang datar, dan harga fungsi f dari setiap pasangan koordinat (x, y) disebut intensitas atau level keabuan (*gray level*) dari sebuah gambar. Ketika x , y , dan nilai intensitas f terbatas dengan nilai diskrit, maka gambar tersebut dapat dikatakan sebagai citra digital (Gonzalez et al, 2002).

Sebuah gambar digital terdiri atas sejumlah elemen yang terbatas, masing-masing memiliki nilai dan lokasi tertentu. Piksel adalah istilah yang paling banyak digunakan untuk menunjukkan elemen digital (Gonzalez et al, 2002). Sebuah citra digital adalah sebuah matriks (array dua dimensi) dari kumpulan piksel sebagai

balok-balok bangunan dasar. Nilai setiap piksel sebanding dengan kecerahan titik yang sesuai dengan lokasinya. Matriks piksel dari sebuah gambar biasanya berbentuk persegi dan digambarkan sebagai piksel $N \times M$ dimana terdiri atas N kolom dan M baris.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2.1 Representasi citra dalam matriks

Sebuah citra seperti sebuah *grid* dengan masing-masing kotak persegi di dalam *grid* berisi satu warna atau piksel. Sebuah citra 8 dengan resolusi 1024x768 adalah sebuah grid dengan 1024 kolom dan 768 baris, yang mana berisi 1024x768 = 786432 piksel. Banyaknya piksel pada sebuah citra tidak menunjukkan dimensi fisik dari sebuah citra.

2.2.3 Pengertian Pengolahan Citra Digital (*Image Processing*)

Sebuah gambar disebut dengan citra digital apabila gambar yang dihasilkan dari proses sebuah komputer, kamera, scanner atau perangkat elektronik lainnya. Pengolahan citra digital diproses oleh komputer dengan menggunakan algoritma. Citra digital direpresentasikan dengan matriks, sehingga pengolahan pada citra digital pada dasarnya memanipulasi elemen-elemen matriks yang berupa piksel (A'la, 2016).

Tujuan dari pengolahan citra antara lain dapat memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik).

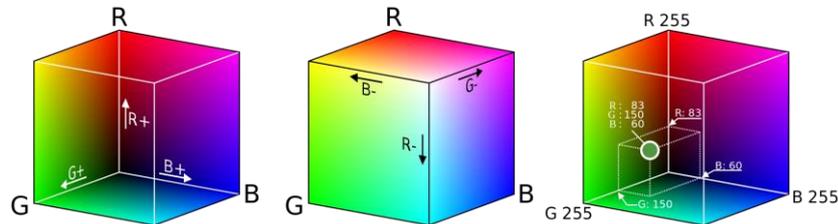
Selain itu juga pengolahan citra dilakukan untuk proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra (Hermawati, 2013).

2.2.4 Citra Warna (RGB)

Citra warna atau biasa dikenal dengan citra RGB merupakan suatu model warna aditif yang terdiri atas tiga buah warna dasar yaitu merah (*red*), hijau (*green*), dan biru (*blue*). Sedangkan warna lain merupakan hasil kombinasi dari ketiga warna tersebut. Model warna RGB merupakan warna dasar yang dapat dibedakan oleh sel kerucut mata manusia berdasarkan teori trikromatik.

Citra warna merupakan penumpukan dari tiga matriks, dengan masing-masing matriks merepresentasikan nilai masing-masing ketiga warna dasar merah, hijau, dan biru dalam setiap piksel sehingga setiap piksel berkaitan dengan tiga nilai (Sianipar, 2013). Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari ketiga warna dasar RGB dengan mengubah nilai HSV (*Hue, Saturation, Values*) agar dapat menghasilkan bermacam-macam warna. Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan kombinasi warna yang bisa disajikan mencapai $2^8 \times 2^8 \times 2^8 = 2^{24}$ atau sekitar 16 juta lebih warna. Untuk sebuah warna dapat direpresentasikan sebagai *vector* tiga dimensi. Misalkan *vector* sebuah warna ditulis sebagai $r = (x, y, z)$, kemudian komponen x , y , dan z digantikan oleh Merah (*Red*), Hijau (*Green*), dan Biru (*Blue*). Sebagai contoh pada Gambar 2.2 sebuah warna ditulis sebagai RGB (83, 150, 60) di mana warna tersebut memiliki nilai $R = 83$, $G = 150$, dan $B = 60$. Sehingga warna tersebut

dilihat oleh mata lebih cenderung ke warna hijau. Sedangkan warna hitam ditulis sebagai RGB (0, 0, 0) dan putih ditulis sebagai RGB (255, 255, 255).

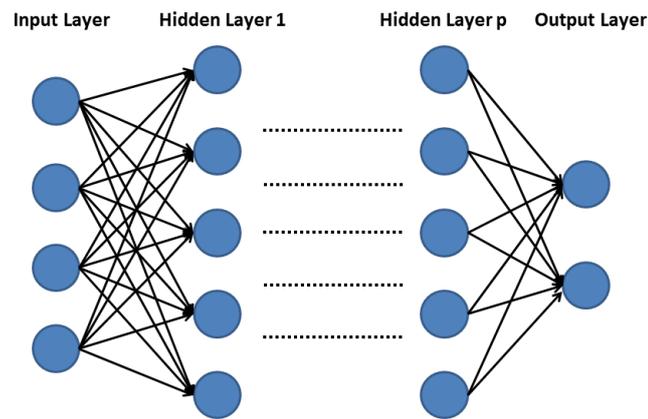


Gambar 2.2 Ruang warna

2.2.5 Deep Learning

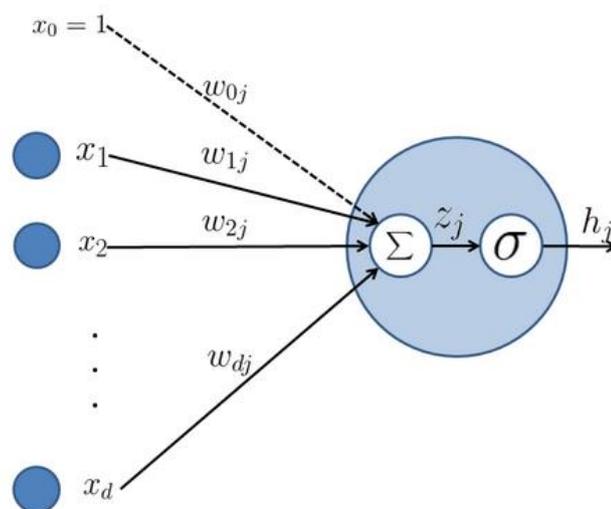
Deep Learning adalah cabang ilmu dari *Machine Learning* berbasis jaringan saraf tiruan (JST) atau bisa dikatakan perkembangan dari JST yang mengajarkan komputer untuk melakukan suatu tindakan yang dianggap alami oleh manusia, misalnya belajar dari contoh. Dalam *deep learning*, sebuah komputer belajar mengklasifikasi secara langsung dari gambar, teks, atau suara. Sebagaimana sebuah komputer dilatih menggunakan data set berlabel dalam jumlah besar lalu kemudian mengubah nilai piksel dari sebuah gambar menjadi representasi internal atau *feature vector* dimana pengklasifikasi dapat mendeteksi atau mengklasifikasi pola pada *input* (LeCun et al, 2015).

Metode deep learning merupakan metode pembelajaran representasi dengan beberapa tingkat representasi, dimana representasi membentuk medan arsitektur jaringan syaraf yang berisi banyak *layer* (lapisan). Lapisan pada *deep learning* terdiri atas tiga bagian, yaitu *input layer*, *hidden layer*, dan *output layer*. Pada *hidden layer* dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar meminimalisir *error* pada *output*.



Gambar 2.3 Layer-layer pada Deep Learning

Gambar 2.3 mengilustrasikan *layer-layer* pada *deep learning* yang memiliki $p + 2$ *layer* (p *hidden layer*, 1 *input* dan 1 *output layer*). Bulatan berwarna biru menggambarkan *neuron*. Disetiap lapis *hidden layer* terdapat satu atau lebih neuron. Neuron-neuron tersebut akan terhubung langsung dengan neuron lain pada *layer* selanjutnya. Koneksi antar neuron hanya terjadi di antara 2 buah *layer* (*input* dan *output*) tidak ada koneksi pada *layer* yang sama walaupun secara teknis bisa saja dibuat dan juga *fully connected*.



Gambar 2.4 Sebuah perceptron dengan d buah input

Sebuah sistem yang terdiri dari sebuah neuron beserta *input* dan *output* nya disebut sebagai *perceptron*. Seperti yang dilihat pada gambar 2.4 sebuah *neuron* diperbesar disimbolkan sebagai h_j yang menerima d buah input $x_1, \dots, x_d \in R$ dapat berasal dari data ataupun *ouput* dari lapisan sebelumnya. Sedangkan x_0 tidak dianggap sebagai *input* (atau sebagai *dummy*) dan selalu bernilai 1. Variabel-variabel $w_{1j}, \dots, w_{dj} \in R$ merupakan bobot/*weights* dari koneksi input ke *neuron* h_j dan w_{0j} dinamakan sebagai bias.

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 bobot inilah yang diatur oleh mesin untuk mengurangi kesalahan yang terjadi. Dalam sistem *deep learning* kemungkinan terdapat ratusan juta bobot yang dapat diatur. Untuk menyesuaikan vektor bobot dengan benar, algoritma menghitung vektor gradien untuk setiap bobot berdasarkan jumlah kesalahan yang meningkat atau menurun jika bobot meningkat dalam jumlah kecil (LeCun et al, 2015).

Operasi pada sebuah *perceptron* merupakan dua buah operasi terpisah yaitu operasi *linear* \sum (2.1) dan operasi *non-linear*/aktivasi σ (2.2). Operasi *linear* dengan nilai bobot yang ada, kemudian hasil komputasi dari operasi *linear* akan ditransformasi menggunakan operasi *non-linear* yang disebut sebagai fungsi aktivasi.

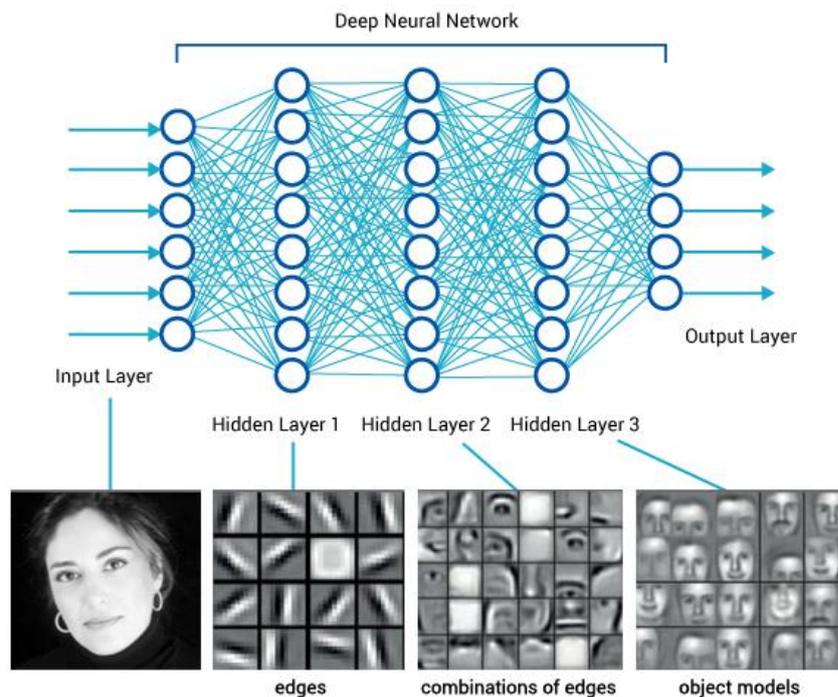
$$z_j = \sum_{i=0}^d w_{ij} x_i \dots\dots\dots (2.1)$$

$$h_j = \sigma(z_j) \dots\dots\dots (2.2)$$

Pada operasi aktivasi/*non-linear* $\sigma : R \rightarrow R$ terdapat berbagai macam fungsi aktivasi yang dapat diimplementasikan pada *hidden neuron*. Di dalam penelitian ini digunakan salah satu fungsi aktivasi yaitu *softmax layer*. *Softmax layer* digunakan apabila pada permasalahan *multiclass classification*, *output layer* biasanya memiliki lebih dari satu *neuron*. Misalkan $a = [a_1, \dots, a_m]^T$ merupakan sebuah vektor dengan m buah elemen, *softmax* didefinisikan sebagai berikut:

$$\sigma(a_j) = \frac{\exp(a_j)}{\sum_{k=1}^m \exp(a_k)} \dots\dots\dots (2.3)$$

Dapat dicek bahwa $\sum_{j=1}^m \sigma(a_j) = 1$ untuk *softmax*.



Gambar 2.5 *Visual object recognition pada deep learning*

Deep learning memungkinkan model komputasi yang terdiri dari beberapa *processing layer* untuk mempelajari representasi data dengan berbagai tingkat

abstraksi. Metode ini telah secara dramatis memperbaiki *state-of-the-art* dalam pengenalan suara (*speech recognition*), pengenalan objek visual (*visual object recognition*), deteksi objek (*object detection*) dan banyak domain lainnya seperti penemuan obat dan genomik. *Deep learning* menemukan struktur yang rumit dalam kumpulan data yang besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana sebuah mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi pada setiap lapisan dari representasi pada lapisan sebelumnya (LeCun, Bengio, & Hinton, 2015).

Beberapa algoritma yang menerapkan konsep *deep learning* antara lain *Deep Convolutional Neural Networks* (DCNN) untuk klasifikasi gambar, *Deep Belief Network – Deep Neural Network* (DBN-DNN) untuk pengenalan suara, *Recurrent Neural Network* (RNN) untuk penerjemahan bahasa, *Query-Oriented Deep Extraction* (QODE) yang berbasis *Restricted Boltzmann Machine* (RBM) untuk peringkasan multi dokumen, *Conditional Restricted Boltzmann Machine* (RBM) untuk memprediksi *Drug-Target Interaction* (DTI), dan *Deep Belief Network* (DBN) untuk prediksi data sesuai waktu.

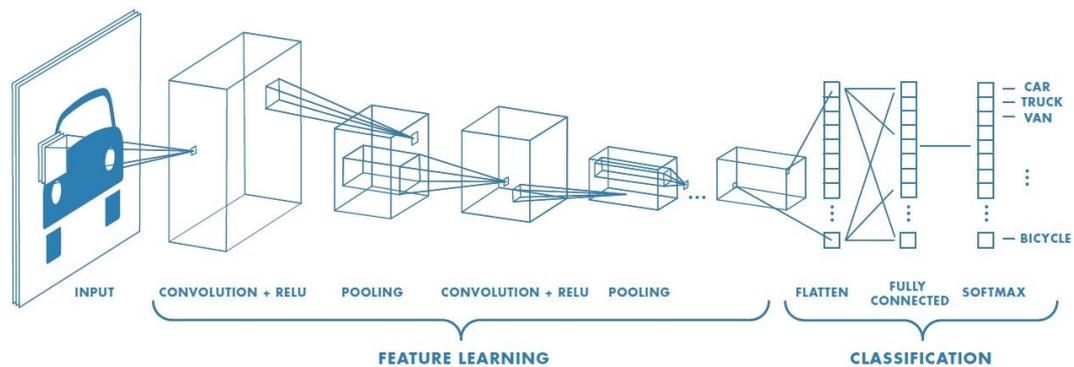
2.2.6 Convolutional Neural Network

Convolutional Neural Network (CNN/ConvNet) adalah salah satu algoritma dari *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN digunakan untuk mengklasifikasi data yang terlabel dengan menggunakan metode *supervised learning*, yang mana cara kerja dari *supervised learning* adalah terdapat data yang dilatih dan terdapat variabel yang

ditargetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada.

CNN sering digunakan untuk mengenali benda atau pemandangan, dan melakukan deteksi dan segmentasi objek. CNN belajar langsung dari data citra, sehingga menghilangkan ekstraksi ciri secara manual. Penelitian awal yang mendasari penemuan ini pertama kali dilakukan oleh Hubel dan Wiesel yang melakukan penelitian *visual cortex* pada indera penglihatan kucing. *Visual cortex* pada hewan sangat *powerful* dalam sistem pemrosesan visual yang pernah ada. Hingga banyak penelitian yang terinspirasi dari cara kerjanya dan menghasilkan model-model baru diantaranya seperti Neocognitron, HMAX, LeNet-5, dan AlexNet.

Lapisan-lapisan CNN memiliki susunan neuron 3 dimensi (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran lapisan sedangkan kedalaman mengacu pada jumlah lapisan. Sebuah CNN dapat memiliki puluhan hingga ratusan lapisan yang masing-masing belajar mendeteksi berbagai gambar. Pengolahan citra diterapkan pada setiap citra latih pada resolusi yang berbeda, dan *output* dari masing-masing gambar yang diolah dan digunakan sebagai *input* ke lapisan berikutnya (gambar 2.6). Pengolahan citra dapat dimulai sebagai fitur yang sangat sederhana, seperti kecerahan dan tepi atau meningkatkan kompleksitas pada fitur yang secara unik menentukan objek sesuai ketebalan lapisan (Mathworks, 2017).



Gambar 2.6 Contoh CNN dengan banyak lapisan

Secara umum tipe lapisan pada CNN dibagi menjadi dua yaitu:

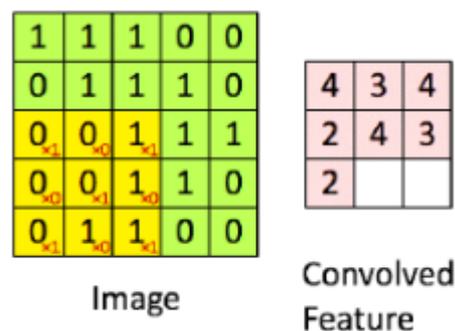
- a) Layer ekstraksi fitur (*feature extraction layer*) gambar, letaknya berada pada awal arsitektur tersusun atas beberapa lapisan dan setiap lapisan tersusun atas *neuron* yang terkoneksi pada daerah lokal (*local region*) dari lapisan sebelumnya. Lapisan jenis pertama adalah *convolutional layer* dan lapisan kedua adalah *pooling layer*. Setiap lapisan diberlakukan fungsi aktivasi dengan posisinya yang berselang-seling antara jenis pertama dengan jenis kedua. Lapisan ini menerima *input* gambar secara langsung dan memprosesnya hingga menghasilkan *output* berupa vektor untuk diolah pada lapisan berikutnya.
- b) Layer klasifikasi (*classification layer*), tersusun atas beberapa lapisan dan setiap lapisan tersusun atas *neuron* yang terkoneksi secara penuh (*fully connected*) dengan lapisan lainnya. Layer ini menerima *input* dari hasil keluaran layer ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa akurasi kelas untuk klasifikasi.

Dengan demikian CNN merupakan metode untuk mentransformasikan gambar asli lapisan per lapisan dari nilai piksel gambar kedalam nilai skoring kelas untuk klasifikasi. Dan setiap lapisan ada yang memiliki *hyperparameter* dan ada yang tidak memiliki parameter (bobot dan bias pada neuron).

Pada penelitian ini terdapat empat macam lapisan utama antara lain:

1) *Convolution Layer*

Convolution Layer melakukan operasi konvolusi pada *output* dari lapisan 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 2.7.



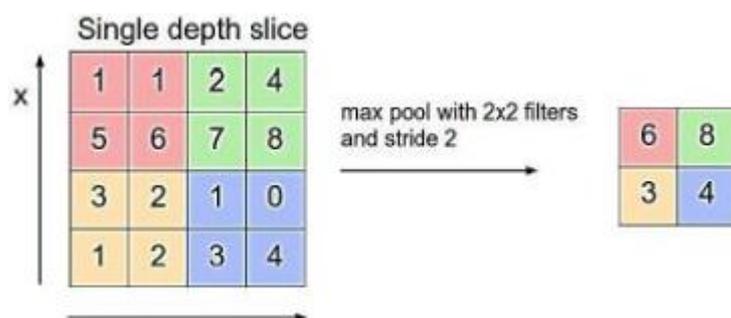
Gambar 2.7 Operasi pada konvolusi

Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. *Kernel* bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi *linear* dari data

input sesuai informasi spasial pada data. Bobot pada lapisan tersebut menspesifikasikan *kernel* konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan *input* pada CNN.

2) *Subsampling Layer*

Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *subsampling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* membagi output dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.8.



Gambar 2.8 Operasi pada *max pooling*

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).

Penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah

convolution layer dengan *stride* yang sama dengan *pooling layer* yang bersangkutan (Springenberg et al, 2014).

3) *Fully Connected Layer*

Layer tersebut adalah lapisan yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan.

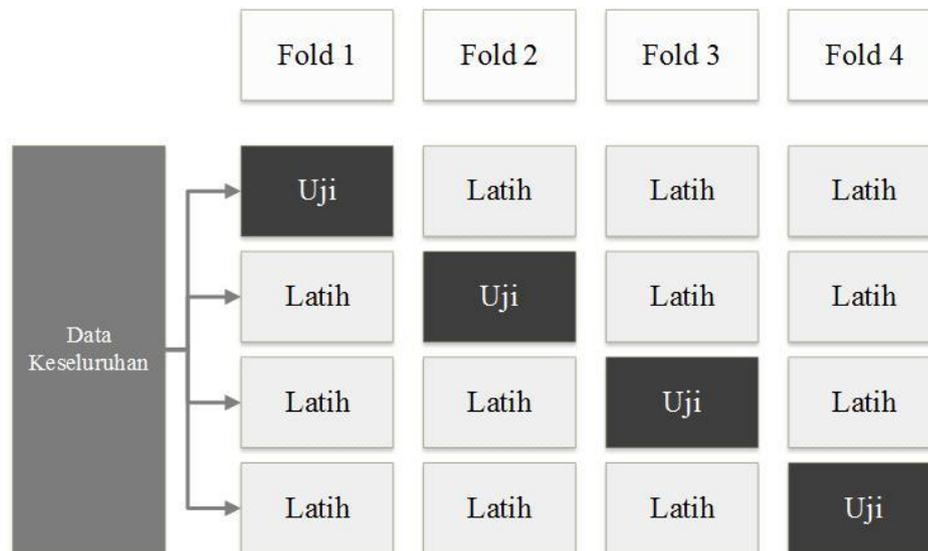
4) Fungsi Aktivasi

Pada penelitian ini fungsi aktivasi yang digunakan adalah *softmax layer*. *Softmax layer* digunakan apabila pada permasalahan *multiclass classification*, *output layer* biasanya memiliki lebih dari satu *neuron*. *Softmax layer* pada dasarnya adalah probabilitas eksponensial yang dinormalisasi dari pengamatan kelas yang diwakili sebagai aktivasi *neuron*. Fungsi eksponensial akan meningkatkan probabilitas nilai maksimum lapisan sebelumnya dibandingkan dengan nilai lainnya.

2.2.7 K-Fold Cross Validation

K-Fold Cross Validation adalah salah satu teknik untuk menilai keakuratan sebuah model yang dibangun berdasarkan dataset yang telah ditentukan. Tujuan dari pembuatan model biasanya untuk melakukan perhitungan maupun klasifikasi terhadap suatu data baru yang kemungkinan belum pernah muncul di dalam

dataset. Dalam proses pembangunan data model, data yang digunakan disebut data latih, sedangkan data yang akan digunakan untuk validasi model disebut dengan data uji.



Gambar 2.9 4-Fold Cross Validation

Salah satu metode *cross-validation* yang populer adalah *K-Fold Cross Validation*. Dalam metode ini dataset penelitian dibagi menjadi partisi-partisi sejumlah K-buah secara acak. Kemudian dilakukan sejumlah K-kali eksperimen, dengan setiap eksperimen menggunakan data partisi ke-K sebagai data uji dan memanfaatkan sisa partisi-partisi lainnya sebagai data latih. Sebagai gambaran *4-Fold Cross Validation* ditunjukkan pada gambar 2.9. Nilai akurasi dari hasil eksperimen yang dilakukan didapatkan dengan mengambil nilai rata-rata dari seluruh eksperimen tersebut.