

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Penelitian terdahulu merupakan suatu yang penting supaya dapat diketahui hubungan antara penelitian yang dilakukan penulis dengan penelitian-penelitian terdahulu. Penelitian yang berjudul “Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network” yang dilakukan Abdul Jalil Rozaqi dkk melakukan klasifikasi penyakit daun kentang dengan menggunakan 3 kelas yaitu: *late blight*, *early blight*, dan daun sehat atau normal. *Dataset* yang digunakan merupakan *dataset* yang didapatkan dari situs Kaggle dengan nama *PlantVillage Dataset*. Data berjumlah 1152 yang terbagi menjadi 500 data pada kelas *late blight*, 500 data pada kelas *early blight* dan 152 pada kelas *healthy* yang kemudian dibagi menjadi 80% untuk data *training* dan 20% untuk data *testing*. Dengan menggunakan metode *Convolutional Neural Network* penelitian ini melakukan *training* sebanyak 10 *epoch* dan menghasilkan akurasi 94% [9].

Penelitian yang dilakukan oleh Santoso dan Ariyanto menggunakan CNN untuk melakukan pengenalan wajah. Pada penelitian ini *dataset* yang digunakan bersumber dari Dr Libor Spacek, dengan hanya mengambil 10 subjek pria yang memiliki 20 gambar tiap subjek yang dibagi menjadi 17 data *train* dan 3 data *testing* sehingga total *dataset* yang digunakan adalah 200 gambar. Pada penelitian ini dilakukan pengujian dengan 4 skenario untuk melihat pengaruh dari kedalaman *layer* dan ukuran gambar data *training* terhadap performa, jumlah kedalaman *layer* yang diuji pada penelitian ini adalah 5 lapisan dan 7 lapisan dan ukuran data *training* 28 x 28 piksel dan 64 x 64 piksel. Dari 4 pengujian didapatkan akurasi paling baik 98,70% pada kedalaman *layer* 7 lapisan dengan ukuran data *training* 28 x 28 piksel [12].

Penelitian lain, yang berjudul “Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks” yang diteliti oleh Limantoro dkk, menggunakan CNN untuk mengklasifikasikan gambar ke dalam dua kelas yaitu kelas motor, dan kelas bukan motor. *Dataset* didapatkan dari pengambilan video yang diletakkan pada dashboard mobil menghadap ke depan yang kemudian akan dilakukan ekstraksi *frame* dan dilakukan *cropping* pada tiap gambar, sehingga total *dataset* yang digunakan adalah 1000 gambar. Pada penelitian ini dilakukan perbandingan performa dari 3 arsitektur CNN yaitu: kecil, sedang, dan besar. Pada arsitektur kecil akurasi yang didapatkan adalah 91,5%, arsitektur sedang 92% dan arsitektur besar 93%, sehingga dapat disimpulkan bahwa arsitektur besar memiliki akurasi yang paling baik [14].

Widiyanto, S dkk pada penelitiannya yang berjudul “Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves” pada *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019* melakukan klasifikasi penyakit pada daun tanaman tomat menggunakan *Convolutional Neural Network*(CNN). Pada penelitian ini dilakukan klasifikasi terhadap 5 kelas yaitu: *healthy, late blight, septoria leaf spot, mozaic virus, tomato yellow leaf curl virus*. *Dataset* yang digunakan sebanyak 1000 gambar pada tiap kelas sehingga total *dataset* berjumlah 5000 gambar yang dibagi 4000 untuk data *training* dan 1000 untuk data *testing*. Penelitian ini menghasilkan nilai akurasi untuk klasifikasi penyakit pada tanaman tomat sebesar 96,60% dan dengan menggunakan *confusion matrix* dapat dilakukan pengukuran *accuracy, precision, recall* dan *F1-Score* dengan nilai rata-rata 98% [10].

Metode *Support Vector Machine* digunakan oleh Rahmawati dkk untuk melakukan klasifikasi pada penyakit daun kentang. Pada penelitian ini terdapat 3 kelas yaitu daun dengan penyakit hawar atau busuk daun atau dalam bahasa inggris disebut *late blight*, penyakit bercak kering atau dalam bahasa inggris disebut *early blight*, dan daun yang sehat. Metode ekstraksi

fitur yang digunakan pada penelitian ini adalah GLCM (*Grey Level Co-occurrence Matrix*) dan *Color Moments* dari masing–masing channel HSV. *Dataset* yang digunakan berasal dari Badan Pengkajian Teknologi Pertanian dengan jumlah data *train* pada masing-masing kelas adalah 100 gambar dan jumlah data *testing* sebanyak 30 gambar pada masing-masing kelas. Pengujian dilakukan dengan mengklasifikasikan seluruh data *testing* yang kemudian akurasi didapatkan dari klasifikasi benar dibagi dengan total data uji, pengujian ini menghasilkan hasil rata-rata akurasi 87% [1].

Dalam penelitian yang berjudul “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network” yang dilakukan oleh S. Ilahiyah dan A.Nilogiri menggunakan arsitektur CNN AlexNet untuk melakukan klasifikasi citra daun untuk mengidentifikasi jenis tumbuhan. Pada penelitian ini pengujian menggunakan *K-Fold Cross Validation* dengan jumlah 10 *fold*. Data yang digunakan berjumlah 2000 citra dari 20 kelas genus tumbuhan yang dibagi menjadi 1800 (90%) data *training* dan 200 (10%) data *testing*. dari pengujian *K-Fold Cross Validation* tersebut didapatkan akurasi rata-rata sebesar 85,2%. Percobaan kedua dilakukan dengan menggunakan 40 data uji yang kemudian dilakukan pengujian sebanyak 3 kali percobaan dan didapatkan rata-rata akurasi 90,8% [16].

Penelitian berjudul “Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca” yang dilakukan Naufal, M membandingkan SVM, KNN dan CNN dalam klasifikasi citra cuaca yang didapatkan dari *Multi-class Weather dataset for image classification* pada repositori *Mendeley* yang memiliki 4 kelas yaitu: *cloudy*, *rain*, *shine*, dan *sunrise* dengan total data 1120 data citra. Proses *training* menggunakan 80% data secara acak, dan proses *testing* menggunakan 20% data. Performa diukur menggunakan *accuracy*, *precision*, *recall*, dan *f1-score* yang dirata-rata menjadi nilai *average performance*. Hasil KNN yang paling baik didapatkan menggunakan jumlah *neighbors* 5 dengan *Euclidean distance* dengan nilai

average performance yang didapatkan 0,766, hasil SVM paling baik menggunakan *kernel* RBF dengan *average performance* yang didapatkan 0,860, hasil CNN paling baik didapatkan *average performance* 0,942. Dengan demikian, dapat disimpulkan CNN memiliki peforma paling baik dengan nilai *accuracy* sebesar 0,942, *precision* sebesar 0,943, *recall* 0,942 dan *f1-score* 0,942 dibandingkan dengan SVM dan KNN [17].

Penelitian berjudul “Identifikasi Penyakit Daun Kentang Berdasarkan fitur warna, tekstur, dan bentuk dengan SVM dan KNN” pada *Seminar Nasional Multimedia & Artificial Intelligence 2020* yang dilakukan Huda dan Nafi’iyah menggunakan metode SVM dan KNN untuk melakukan klasifikasi penyakit pada daun kentang. Fitur yang digunakan pada penelitian ini adalah fitur warna berupa nilai rata-rata RGB, standar deviasi RGB, *skewness* RGB, dan *entropy* RGB, fitur tekstur berupa nilai rata-rata *grayscale*, standar deviasi *grayscale*, dan *Gray Level Co-Occurance Matrix*, dan fitur bentuk berupa *area*, *perimeter*, *metric*, *major axis*, *minor axis*, dan *eccentricity*. *Dataset* yang digunakan pada penelitian ini berjumlah 1200 data yang dibagi menjadi 900 data *training* dan 300 data *testing*. Hasil penelitian ini menunjukkan dengan algoritma SVM akurasi paling baik dengan menggunakan fitur warna 88%, dan dengan algoritma KNN dengan $K = 5$ akurasi paling baik dengan menggunakan fitur warna 85% [8].

Penelitian dengan judul “Implementasi Convolutional Neural Network Untuk Identifikasi Ikan Air Tawar” yang di teliti oleh S. Fauzi dkk melakukan klasifikasi 10 jenis ikan air tawar menggunakan CNN. Data yang digunakan berjumlah 300 gambar yang terdiri dari 10 kelas yaitu: ikan Gurame Padang, Mas Kaca, Mas Orange, Mas Putih, Mujair, Nila, Patin, Bawal, Belut. Dan Lele. *Dataset* dibagi menjadi 80% data *training* dan 20% data *testing*. Hasil akurasi yang didapatkan pada penelitian ini adalah 88,33% [18].

Penelitian yang dilakukan E. Nitisih dkk dengan judul “Tomato Leaf Disease Detection using Convolutional Neural Network with Data Augmentation” melakukan klasifikasi pada 5 penyakit daun tomat yaitu:

Bacterial spot, Early blight, Septoria leaf spot, Tomato yellow leaf curl virus, Tomato mosaic virus. Dataset yang digunakan berasal dari dataset *PlantVillage* yang berjumlah 12206 gambar. Data dibagi menjadi 80% untuk data *training* dan 20% untuk data *testing*. Pada penelitian ini data *training* dilakukan augmentasi data dari 9801 data menjadi 39204 atau 4 kali lebih besar dari jumlah awal. Proses *training* dilakukan pada data yang belum diaugmentasi menghasilkan akurasi sebesar 94,61% dengan menggunakan *confusion matrix* dan 97,01% pada data yang sudah dilakukan augmentasi data [11].

Tabel 2.1 Penelitian Sebelumnya

No	Penulis	Tahun	Judul	Metode	Persamaan	Perbedaan	Kesimpulan
1	A. Rozaqi, A. Sunyoto, M. Arief	2021	Deteksi Penyakit Pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode Convolutional Neural Network	CNN	Metode yang digunakan sama	Arsitektur yang digunakan berbeda	Penelitian ini berhasil menerapkan CNN untuk klasifikasi 3 kelas pada daun kentang dengan hasil akurasi sebesar 94%
2	A. Santoso, G. Ariyanto	2018	Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah	CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	Penelitian ini berhasil menerapkan CNN untuk pengenalan wajah dengan akurasi 98,75%. Pada penelitian ini dapat ditarik kesimpulan bahwa semakin banyak layer digunakan maka semakin baik hasil yang didapatkan
3	S. Limantoro, Y. Kristian, D. Purwanto	2017	Deteksi Pengendara Sepeda Motor Menggunakan Deep Convolutional Neural Networks	CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	Pada arsitektur kecil akurasi yang didapatkan adalah 91,5%, arsitektur sedang 92% dan arsitektur besar 93%, sehingga dapat disimpulkan bahwa arsitektur besar memiliki akurasi yang paling baik
4	S. Widiyanto, R. Fitrianto, D. Wardani	2019	Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves	CNN, Data Augmentation	Metode yang digunakan sama	Objek yang diteliti berbeda	Penelitian ini menghasilkan nilai akurasi untuk klasifikasi penyakit pada tanaman tomat sebesar 96,60% dan dengan menggunakan <i>confusion matrix</i> dapat dilakukan pengukuran <i>accuracy</i> , <i>precision</i> , <i>recall</i> dan <i>F1-Score</i> dengan nilai rata-rata 98%
5	P. Rakhmawati, Y. Pranoto, E. Setyati	2018	Klasifikasi Penyakit Daun Kentang Berdasarkan Fitur Tekstur Dan Warna Menggunakan Support Vector Machine	SVM, GLCM, Color Moment	Objek yang diteliti sama yaitu penyakit pada daun kentang	Metode yang digunakan berbeda	Penelitian ini berhasil melakukan klasifikasi pada penyakit daun kentang menggunakan metode multi SVM kernel RBF dengan akurasi 87%

No	Penulis	Tahun	Judul	Metode	Persamaan	Perbedaan	Kesimpulan
6	S. Ilahiyah, A. Nilogiri	2018	Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network	CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	Penelitian ini menggunakan <i>K-Fold Cross Validation</i> dan mendapatkan akurasi sebesar 85,2%. Dan pengujian menggunakan data uji baru mendapatkan rata-rata akurasi 90,8%.
7	M. Naufal	2021	Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca	SVM, KNN, dan CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	CNN memiliki performa paling baik dengan nilai accuracy sebesar 0,942, precision sebesar 0,943, recall 0,942 dan f1-score 0,942 dibandingkan dengan SVM dan KNN
8	M. Huda, N. Nafi'iyah	2020	Identifikasi Penyakit Daun Kentang Berdasarkan Fitur Warna, Tekstur, dan Bentuk dengan SVM dan KNN	SVM dan KNN GLCM	Objek yang diteliti sama	Metode yang digunakan berbeda	Penelitian ini berhasil melakukan identifikasi penyakit daun kentang dengan menggunakan fitur warna menggunakan metode SVM dengan akurasi 88% dan dengan metode KNN dengan K = 5 dengan akurasi 85%
9	S. Fauzi, P. Eosina, G. Laxmi	2019	Implementasi Convolutional Neural Network Untuk Identifikasi Ikan Air Tawar	CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	Dalam melakukan klasifikasi terhadap 10 jenis ikan air tawar menggunakan CNN, penelitian ini mendapatkan akurasi 88,37%
10	E. Nithish, M. Kaushik, P. Prakasih, dkk	2020	Tomato Leaf Disease Detection using Convolutional Neural Network with Data Augmentation	CNN	Metode yang digunakan sama	Objek yang diteliti berbeda	Penelitian ini membandingkan hasil akurasi CNN pada training yang menggunakan data augmentasi dan tidak menggunakan data augmentasi. Hasil penelitian ini adalah akurasi sebesar 94.61% pada data yang tidak di augmentasi dan 97,01% pada data yang diaugmentasi.

2.2 Landasan Teori

Pada subbab ini akan dibahas beberapa landasan teori yang berkaitan dengan penelitian ini. Berikut adalah beberapa landasan teori yang berkaitan dengan penelitian ini.

2.2.1 Bercak Kering dan Hawar Daun Kentang

Kentang atau *Solanum tuberosum L* merupakan salah satu tanaman yang paling berharga dan sangat penting dari segi produksi maupun konsumsi di dunia karena memiliki nutrisi yang tinggi. Produksi kentang berada di peringkat keempat setelah gandum (*Triticum aestivum L.*), padi (*Oryza sativa L.*), dan jagung (*Zea mays L.*) [5], [7], [19]. Di Indonesia sendiri, kentang banyak ditanam dan tumbuh pada ketinggian 1000 – 2000 meter di atas permukaan laut [20], [21]. Seperti yang sudah dijelaskan pada bagian latar belakang bahwa produksi kentang dan konsumsi kentang oleh rumah tangga mengalami kenaikan dari beberapa tahun terakhir [2]. Kentang memegang peranan penting karena bernilai ekonomi tinggi dan mempunyai potensi untuk mendukung diversifikasi pangan karena kentang merupakan sumber karbohidrat yang bermanfaat untuk meningkatkan energi dalam tubuh dan memiliki protein yang baik [3], [7].

Salah satu penyakit yang sering dijumpai pada daun tanaman kentang adalah penyakit bercak kering (*early blight*) [1]. Penyakit ini disebabkan oleh *Alternaria Solani*. Penyakit ini menyerang pada berbagai kondisi iklim dan dapat menyebabkan kerusakan parah jika tidak terkendali. Penyakit ini merupakan penyakit *poly cyclic* yang artinya dapat menyerang tanaman lebih dari satu kali dalam satu musim tanam. Penyakit ini biasanya menyerang daun yang sudah lebih matang atau tua ketimbang daun yang lebih muda dengan ciri-ciri bercak yang berbentuk seperti *bull's eye* atau mata banteng yang berwarna abu-abu hingga coklat pada bagian tengah. Kehilangan hasil tanam yang disebabkan oleh penyakit ini bervariasi dari 5% hingga 50%. Menerapkan teknik penanaman yang baik dan pemberian fungisida merupakan hal yang penting untuk membantu pengendalian penyakit bercak

kering pada tanaman kentang. Gambar 2.1 dan Gambar 2.2 adalah contoh daun yang terserang penyakit bercak kering [4], [5].



Gambar 2.1 Daun yang Terserang Penyakit Bercak Kering [4]



Gambar 2.2 Daun yang Terserang Penyakit Bercak Kering [22]

Selain penyakit bercak kering, penyakit hawar atau busuk daun (*late blight*) juga sering dijumpai pada tanaman kentang [1]. Penyakit ini disebabkan oleh patogen *Phytophthora Infestans* yang menyebar melalui air, angin atau serangga dan sangat dipengaruhi oleh suhu, kelembaban, dan curah hujan. Penyakit ini biasanya menyerang tanaman kentang yang berada pada dataran tinggi, memiliki suhu rendah dan juga memiliki kelembaban yang tinggi. Biasanya penyakit ini timbul pada tanaman berumur 5-6 minggu setelah tanam [23]. Kerugian panen akibat penyakit hawar atau busuk daun

bervariasi dari 10% hingga 80%. Penyakit ini juga dinilai mengancam keamanan pangan, kesehatan manusia, lingkungan serta kerugian ekonomi [7].

Gejala awal penyakit ini adalah munculnya bercak basah pada tepi atau ujung daun, dan kemudian akan melebar berwarna cokelat atau abu-abu yang gelap dan basah [23]. Perbedaan gejala penyakit ini dengan penyakit bercak kering adalah penyakit ini memiliki gejala awal bercak kebasah-basahan di tepi daun sedangkan penyakit bercak kering memiliki gejala awal bercak di tengah atau di bagian bawah daun yang berbentuk seperti mata banteng [4], [6], [23].

Pengendalian penyakit ini dapat dilakukan dengan melakukan sanitasi pada bagian tanaman kentang yang terinfeksi, pemilihan bibit yang sehat dan tidak cacat pada saat penanaman, pencelupan bibit sebelum tanam pada larutan *Pseudomonas fluorescens* (Pf) dengan dosis 1ml Pf per liter air selama kurang lebih 3 menit. Penyemprotan fungisida dilakukan untuk menghambat penetrasi dan perkecambahan patogen sehingga jika spora sudah melakukan penetrasi ke dalam daun maka penggunaan fungisida menjadi tidak efektif [23]. Gambar 2.3 merupakan contoh daun kentang yang mengalami penyakit hawar.



Gambar 2.3 Daun yang Terserang Penyakit Hawar [6]

2.2.2 *Image Classification*

Image classification adalah dimana komputer dapat menganalisa gambar dan mengidentifikasi kelas dari gambar tersebut. Kelas merupakan label dari suatu objek misalkan mobil, bangunan, hewan dan lain-lain. Misalkan komputer diberi input sebuah gambar domba, *image classification* merupakan proses dari komputer untuk menganalisa gambar tersebut dan memberitahu bahwa itu adalah domba atau berapa probabilitas bahwa itu adalah domba [24]. *Image classification* juga dapat dikatakan sebagai suatu *task* untuk melakukan pelabelan pada *image* input dengan probabilitas dari kelas-kelas tertentu [25].

Image classification dapat menyelesaikan berbagai masalah dibidang informatika, *agriculture*, kelautan, kesehatan, dan juga bisnis. Dalam bidang *agriculture* sendiri *image classification* dapat mendeteksi penyakit tanaman seperti pada jagung, padi, bunga, sayuran, ketimun, tomat, dan lain sebagainya [10].

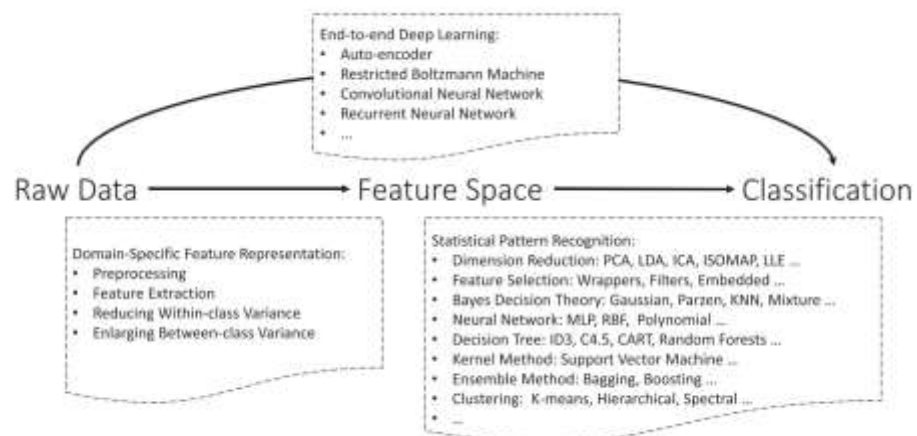
Dalam *image classification* diberikan input gambar-gambar yang terdiri dari beberapa kelas kemudian dilakukan *training classifier* atau model dan kemudian dilakukan evaluasi yaitu membandingkan kelas sebenarnya dari gambar dan kelas yang terprediksi oleh *classifier* atau model yang sudah di *training* [24], [26].

2.2.3 *Pattern Recognition*

Pengenalan pola atau *pattern recognition* merupakan proses mengenali suatu objek dari fitur-fitur yang didapatkan dari suatu objek kemudian dibandingkan dengan fitur-fitur yang telah tersimpan di basis data. Pengenalan pola sudah banyak diimplementasikan dalam berbagai bidang seperti pengenalan sidik jari, pengenalan wajah, pengenalan iris mata dan lain sebagainya [27]. Tujuan dari pengenalan pola adalah untuk mengelompokkan atau mengategorikan suatu objek berdasarkan ciri-ciri pada objek tersebut, atau dengan kata lain pengenalan pola adalah proses untuk membedakan satu objek dengan objek lainnya [28].

Metode dalam pengenalan pola secara umum dibagi menjadi 2 kategori yaitu: *two-stage*, dan *end to end* yang secara singkat dapat dilihat pada Gambar 2.4. Kebanyakan metode-metode konvensional termasuk ke dalam kategori two stage. Pada kategori ini data mentah dirubah ke dalam *feature space* dengan *feature representation*. Dalam tahap *feature representation* dilakukan *preprocessing* seperti menghilangkan *noise* dan menormalisasi data, ekstraksi fitur, mengurangi varians di dalam kelas dan menambah varians antar kelas. Setelah dilakukan *feature representattion*, tahap selanjutnya adalah klasifikasi pola. Tahap ini dikenal juga sebagai *statistical pattern recognition*. Di tahap ini banyak metode yang digunakan seperti reduksi dimensi, seleksi fitur, *bayes decision theory*, *artificial neural network*, *decision tree-based*, *kernel methods* (SVM), dan juga *clustering* [29].

Metode *deep learning* termasuk ke dalam kategori *end to end*. Dimana dalam kategori ini representasi dan klasifikasi fitur dilakukan bersama-sama dari data mentah. Dengan cara ini, fitur dan *classifier* yang telah dilatih lebih fleksibel dan diskriminatif daripada kategori *two stage* [29].



Gambar 2.4 Dua Kategori Umum Pengenalan Pola [29]

2.2.4 Deep Learning

Deep learning adalah bagian dari algoritma *machine learning*, yang bertujuan mempelajari abstraksi tingkat tinggi dalam data dengan

memanfaatkan arsitektur hierarkis. Baru-baru ini, banyak algoritma *deep learning* yang di usulkan untuk mengatasi masalah kecerdasan buatan. *Deep learning* dapat digunakan untuk berbagai permasalahan seperti pengenalan citra atau klasifikasi citra, deteksi objek, *image retrieval*, *semantic segmentation* dan *human pose estimation* [25].

Dalam beberapa tahun terakhir banyak metode-metode terkait *deep learning* muncul. Secara umum, metode-metode ini dibagi menjadi 4 kategori berdasarkan dasar algoritma tersebut : *Convolutional Neural Networks* (CNNs), *Restricted Boltzmann Machines* (RBMs), *Autoencoder and Sparse Coding* [25].

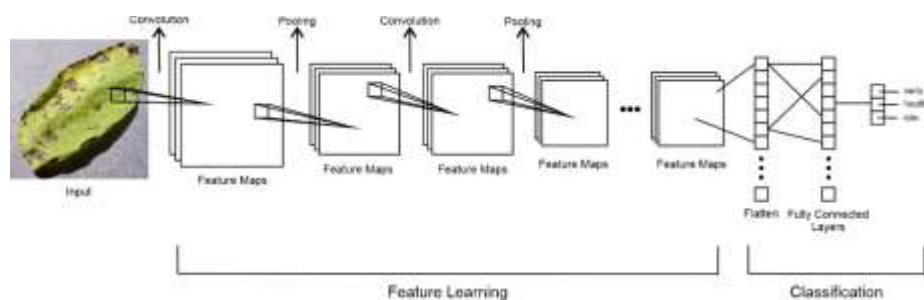
Deep learning kaya akan metode yang meliputi *neural networks*, model probabilitas hirarkis, dan berbagai *unsupervised* dan *supervised learning* lainnya. *Deep learning* menjadi trend saat ini karena terbukti mengungguli metode-metode canggih sebelumnya [30].

2.2.4.1 Convolutional Neural Network

Algoritma *Convolutional Neural Network* (CNN) berawal dari algoritma *Multilayer Perceptron* (MLP) yang kemudian dikembangkan. *Convolutional Neural Network* termasuk ke dalam *deep learning* yang sudah dipakai secara luas yang terinspirasi dari *visual cortex* hewan. *Convolutional Neural Network* ini sering digunakan untuk melakukan *object recognition* (pengenalan objek) [13], [31].

Algoritma ini diperuntukkan untuk mengolah data 2 dimensi dan merupakan salah satu *deep learning* yang dapat digunakan untuk pengenalan citra digital [18], [31]. Dengan menggunakan Jaringan Syaraf Tiruan biasa pemilihan fitur dilakukan secara manual sehingga harus dilakukan dengan sangat teliti. Namun dengan CNN, pemilihan fitur dilakukan secara otomatis dan sudah termasuk di dalam proses *training*. Hal ini merupakan salah satu kelebihan dari CNN [32].

Convolutional Neural Network telah berhasil dalam mengatasi masalah *computer vision* seperti pengenalan wajah, deteksi objek, robotika, dan bahkan *self-driving cars*. *Convolutional Neural Network* terdiri dari 3 jenis *neural layers* utama yaitu: *convolutional layers*, *pooling layers*, dan *fully connected layers* yang masing-masing memiliki peran penting dalam CNN [30]. Dalam melakukan perancangan arsitektur CNN tidak ada cara yang pasti dalam menentukan nilai-nilai parameter dari arsitektur, biasanya menggunakan intuisi dan *trial and error* saja. Namun, dari beberapa arsitektur terdahulu dapat dipelajari pola-pola arsitekturnya sehingga dalam merancang arsitektur terdapat pola-pola yang dapat diikuti sehingga menghasilkan arsitektur yang baik [33]. Dalam melakukan perancangan arsitektur CNN biasanya memiliki pola Conv-Pool-Conv-Pool atau Conv-Conv-Pool-Conv-Conv-Pool [34]. Dari beberapa penelitian dan arsitektur terdahulu ada beberapa input size yang digunakan seperti 28x28, 32x32, 64x64, 128x128, 224x224, 299x299 [10], [12], [35]. Gambaran umum proses CNN dapat dilihat pada Gambar 2.5.

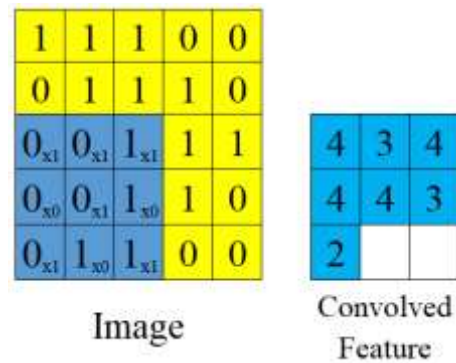


Gambar 2.5 Gambaran Umum CNN

a. *Convolutional layer*

Layer ini merupakan langkah awal dari CNN, *layer* ini mungkin saja terdiri atas beberapa *layer* atau dengan kata lain tidak hanya satu *layer* saja [37]. *Convolution layer* merupakan *layer* pertama yang mengekstraksi fitur dari *input image* [36]. Di *layer* ini, CNN menggunakan bermacam-macam *kernel* untuk melakukan konvolusi terhadap citra *input* yang akan menghasilkan *output* berupa *feature maps* atau *convolved feature*. Proses konvolusi ini merupakan kelebihan dari CNN yaitu memiliki waktu

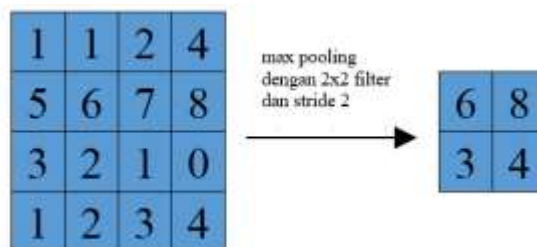
learning yang cepat [30]. Ukuran *convolution layer* yang biasa digunakan adalah bilangan-bilangan ganjil yaitu: 3x3, 5x5, dan 7x7 [38]. Sementara, jumlah *filter* yang biasa digunakan pada tiap-tiap *layer* yaitu: 32–64–128 atau 32–32–64–64, atau dapat dikatakan polanya menaik dua kali lipat [34]. Gambar 2.6 merupakan gambaran dari proses *convolution layer*.



Gambar 2.6 Ilustrasi Proses Convolution Layer

b. Pooling Layers

Pooling layers berguna untuk merubah dimensi dari input ke *layer* konvolusi berikutnya menjadi lebih kecil. Proses ini juga disebut dengan *subsampling* dan *downsampling*. Rata-rata *pooling* yang digunakan adalah MaxPooling dan AveragePooling. Dari beberapa penelitian MaxPooling memiliki performa yang lebih baik dibandingkan AveragePooling [30]. Gambar 2.7 merupakan contoh MaxPooling.



Gambar 2.7 Ilustrasi Max Pooling

c. Stride

Stride merupakan jumlah pergeseran *filter* atas input matriks baik secara *horizontal* maupun *vertical*. Jika *stride* adalah 1 maka *filter* bergeser satu pixel satu pixel, jika *stride* 2 maka *filter* bergeser dua piksel dua piksel

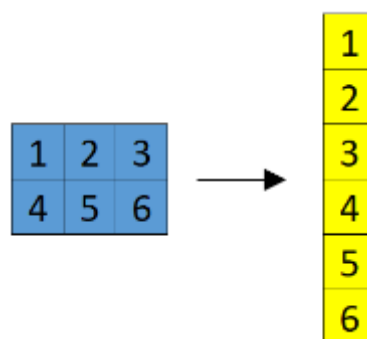
dan seterusnya. Pilihan umum pada *stride* umumnya adalah 1, namun nilai *stride* lebih dari 1 terkadang digunakan untuk mencapai *downsampling* pada *feature maps* [34], [36], [38].

d. *Padding*

Padding merupakan salah satu teknik untuk mempertahankan ukuran *output* sama dengan ukuran *inputnya* dengan cara menambahkan *padding* berupa nilai 0 pada sisi luar dari gambar. Dengan digunakannya *padding* ini dapat mengatasi masalah dimana piksel yang berada di sudut dan tepi tidak terproses sebanyak piksel yang ditengah. Ada dua macam *padding* yang dapat digunakan yaitu *valid* dan *same padding*. *Valid padding* berarti tidak menggunakan *padding* sehingga pada *valid padding* ukuran *output* akan menjadi lebih kecil dibandingkan *inputnya*. *Same padding* berarti menggunakan *padding* sehingga ukuran *output* akan memiliki ukuran yang sama dengan ukuran *inputnya*[39], [40].

e. *Fully Connected Layers*

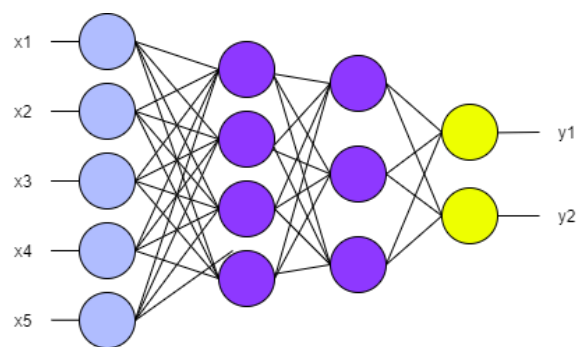
Pada *layer* ini pertama-tama dilakukan *flatten layer* yang merubah *feature maps* 2 dimensi ke dalam 1 dimensi [30]. Hasil dari *flatten layer* ini yang akan dimasukkan ke dalam *fully connected layers* seperti pada *neural network* [18]. Ilustrasi *flatten layer* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Ilustrasi Flatten Layers

Setelah dilakukan *flatten layers*, setiap *input* terhubung pada neuron di *hidden layer* untuk didapatkan hasil klasifikasi berdasarkan nilai bobot dan bias pada setiap neuron yang ada pada *hidden layer* menggunakan

fungsi aktivasi *softmax* ataupun *sigmoid* [18], [36]. Fungsi aktivasi *softmax* digunakan untuk klasifikasi yang memiliki lebih dari dua kelas, sementara *sigmoid* untuk klasifikasi dua kelas. *Layer* ini merupakan *layer* terakhir sehingga setelah *layer* ini tidak ada lagi *convolution layer* [25]. Dalam menentukan jumlah neuron *hidden layers* umumnya jumlahnya berkurang pada tiap *layer* [14], [34]. Ilustrasi *fully connected layer* dapat dilihat pada Gambar 2.9.

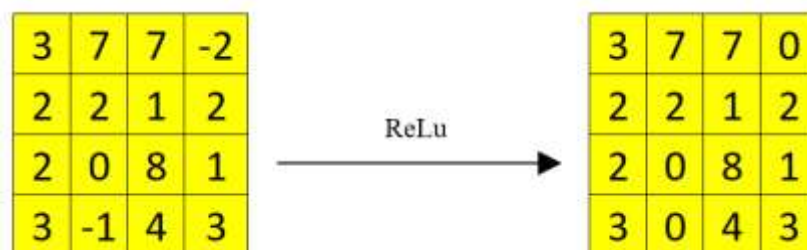


Gambar 2.9 Ilustrasi Fully Connected Layer

f. Fungsi Aktivasi ReLu

Fungsi aktivasi adalah fungsi yang mengambil satu bilangan dan melakukan beberapa operasi matematika padanya. ReLu (*Rectified Linear Unit*) adalah fungsi aktivasi linier yang memiliki *thresholding nol* [13]. Fungsi aktivasi ini mengubah nilai negatif menjadi nol pada hasil *layer* sebelumnya [18]. Adapun persamaan fungsi aktivasi ReLu dapat dilihat pada Persamaan (2.1) [13]. Gambaran proses fungsi aktivasi ReLu dapat dilihat pada Gambar 2.10.

$$\text{rect}(x) = \max(0, x) \quad (2.1)$$



Gambar 2.10 Fungsi Aktivasi ReLu

2.2.5 Confusion Matrix

Confusion matrix mengukur performa klasifikasi pada suatu *classifier* pada beberapa data *test*. *Confusion matrix* berbentuk matriks 2 dimensi dimana dimensi pertama berupa kelas aktual atau *true class* atau *actual class* dari suatu objek sedangkan dimensi kedua merupakan kelas hasil klasifikasi dari *classifier* [41].

Dalam *confusion matrix* ada 4 istilah yang digunakan untuk mengukur performa klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP) yaitu data kelas positif yang terklasifikasi dengan benar, *True Negative* (TN) yaitu data kelas negatif yang terklasifikasi dengan benar, *False Positive* (FP) yaitu data kelas negatif yang terklasifikasi positif, *False Negative* (FN) yaitu data kelas positif yang terklasifikasi negatif [42]. Gambar 2.11 merupakan representasi dari nilai TP, FP, FN dan TN dalam *confusion matrix*.

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Gambar 2.11 Metric pada Confusion Matrix [43]

Penelitian ini akan menggunakan *classification report* yang akan menunjukkan metrik-metrik klasifikasi sebagai berikut [44] :

a. *Accuracy*

Accuracy berarti keakuratan keseluruhan model, yang berarti total sampel yang diklasifikasikan benar oleh model. persamaan (2.2) merupakan persamaan untuk mencari nilai akurasi [42], [43].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \% \quad (2.2)$$

dimana:

TP : True Positive

TN : True Negative

FP: False Positive

FN: False Negative

b. Precision

Precision berarti seberapa presisi model dari data yang sebenarnya positif diantara data-data yang terprediksi positif (*predicted positive*) [42], [45].

$$precision = \frac{TP}{TP + FP} \times 100 \% \quad (2.3)$$

dimana:

TP : True Positive

FP: False Positive

c. Recall

Recall berarti banyaknya data *actual positive* yang terklasifikasi atau terlabeli sebagai *true positive*. Persamaan (2.4) merupakan persamaan untuk mencari nilai recall [42], [45].

$$recall = \frac{TP}{TP + FN} \times 100 \% \quad (2.4)$$

dimana:

TP : True Positive

FN: False Negative

d. F1-Score

F1-Score menggabungkan pengukuran dari *Precision* dan *Recall* menjadi satu pengukuran. *F1-Score* merupakan pengukuran yang lebih baik

jika ada distribusi kelas yang tidak merata. *F1-Score* dapat dihitung menggunakan persamaan (2.5) [45].

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (2.5)$$

dimana:

TP : True Positive

TN : True Negative

FP: False Positive

FN: False Negative

2.2.6 Augmentasi Data

Augmentasi data adalah teknik untuk memanipulasi dan memperbanyak data tanpa kehilangan esensi dan inti dari data tersebut. Dalam *deep learning* dibutuhkan data yang banyak dan bervariasi untuk mendapatkan hasil yang bagus [10]. Augmentasi merupakan salah satu solusi dalam perbanyak data yang dapat berupa *scaling*, *rotating*, dan *cropping* [25]. Pada penelitian sebelumnya, augmentasi dilakukan hanya dengan menerapkan *rotating*, sehingga pada penelitian ini hanya akan dilakukan *rotating* saja [10].

Rotating dilakukan dengan memutar gambar secara *random* dengan derajat tertentu. Apabila gambar yang dirotasi bukan dengan kelipatan 90 derajat maka *augmentor* akan memotong gambar secara otomatis [46]. Gambar 2.12 merupakan salah satu contoh penerapan *rotating* dalam *augmentation*, sedangkan Gambar 2.13 merupakan contoh data daun kentang yang dilakukan *augmentation* dengan teknik *rotating*.



Gambar 2.12 Contoh *rotation* pada *augmentation*



Gambar 2.13 Contoh Augmentasi Data