BAB I
PENDAHULUAN

1.1 Latar Belakang

Sebagai salah satu negara dengan keanekaragaman tanaman anggrek, Indonesia menampilkan *varietas* yang sangat beragam. Berbagai spesies anggrek dapat ditemukan di sejumlah pulau di Indonesia, termasuk Sumatera, Jawa, Kalimantan, Sulawesi, Nusa Tenggara, Bali, Maluku, dan Papua. Setiap pulau menawarkan jenis anggrek dengan karakter bunga yang bervariasi dan unik. Tanaman ini tersebar luas di seluruh penjuru Indonesia, menempati berbagai habitat mulai dari pantai, hutan bakau, padang rumput, hingga puncak gunung. Indonesia juga berperan secara signifikan dalam jumlah total spesies anggrek di dunia, diperkirakan mencapai sekitar 25.000, dengan sekitar 5.000 di antaranya tumbuh secara alami di berbagai wilayah Indonesia. Keberagaman geografis dan kondisi alam di Indonesia menjadi faktor utama yang mendukung keberadaan dan kelimpahan anggrek di berbagai ekosistem (Rizki et al. 2023).

Anggrek merupakan salah satu keluarga tanaman berbunga dengan jumlah *varietas* terbanyak dan paling beragam. Keanekaragaman ini sangat terasa dalam praktik pertanian di Indonesia, di mana berbagai jenis anggrek dibudidayakan secara intensif dan berkelanjutan, menciptakan kekayaan flora yang bernilai tinggi. Namun, keragaman tersebut juga menimbulkan tantangan, karena banyak orang mengalami kesulitan dalam membedakan satu jenis anggrek dengan yang lainnya (Syahputra and Wibowo 2020). Selain itu, perawatan anggrek juga memerlukan perhatian khusus, sebab setiap jenis memiliki kebutuhan yang berbeda agar dapat tumbuh dan berbunga dengan optimal (Husodo, Lubis, and Rusdi 2023). Bunga anggrek hasil persilangan sangat diminati, karena umumnya memiliki karakteristik menarik seperti warna, pola, bentuk, serta ketahanan bunga agar tidak mudah layu—baik sebagai bunga potong maupun bunga dalam pot. Proses

hibridisasi dilakukan dengan memanfaatkan varietas anggrek alami yang berasal dari Indonesia (Yogyakarta, Sugiyarto, and Umniyatie 2016).

Berdasarkan penelitian terdahulu yang berkaitan dengan masalah yang akan dibahas yang berjudul "Klasifikasi Bunga Anggrek untuk *Genus Grammatophyllum* Menggunakan Metode *Convolutional Neural Network* (CNN)" yang ditulis oleh Puspita sari dan Agung Toto Wibowo dari Universitas Telkom Bandung pada Tahun 2021. Penelitian ini membahas mengenai klasifikasi bunga anggrek untuk genus *grammatophyllum* dengan menggunakan metode CNN dengan mengimplementasikan arsitektur *Alex Net* dan *custom* arsitektur CNN. Penelitian ini bertujuan untuk melakukan image recognition terhadap bunga anggrek genus *grammatophyllum* terhadap empat spesies bunga menggunakan algoritma CNN. Hasil dari penelitian ini menunjukan bahwa Performansi terbaik pada arsitektur *Alex Net* didapat dengan mengimplementasikan *dropout* 30% yaitu *precision* 79.76%, *recall* 83.33%, f score 84.10% dan akurasi sebesar 86.07%. Sedangkan untuk *custom* arsitektur CNN didapat dengan hasil implementasi *dropout* 20% yaitu *precision* 93.75%, *recall* 91.66%, *f score* 91.74% dan akurasi sebesar 91.42% (Toto Wibowo 2021).

Berdasarkan data yang diambil di lapangan dan juga hasil wawancara di Anggrek Kuningan dengan Bapak Ujang Moch Furkon selaku pemilik Anggrek Kuningan, Anggrek Kuningan adalah salah satu toko yang menjual sekaligus membudidayakan tanaman anggrek yang berada di Darma, Kuningan, Jawa Barat. Toko ini menjual berbagai tanaman hias anggrek dengan berbagai macam jenis. Adanya kesulitan dalam membedakan jenis-jenis tanaman anggrek menjadi permasalahan yang terjadi di Anggrek Kuningan, sehingga menyulitkan untuk memberikan informasi kepada pelanggan.

Perkembangan teknologi di bidang ilmu pengetahuan, pengolahan citra digital membuka peluang untuk secara otomatis membedakan dan mengidentifikasi jenis varietas anggrek. Klasifikasi citra digital menjadi solusi alternatif untuk mengatasi tantangan tersebut. Tujuan dari klasifikasi

citra ini adalah me replikasi kemampuan manusia dalam memahami informasi citra *digital*, sehingga komputer dapat mengenali objek pada citra dengan cara yang serupa seperti manusia. Namun, proses feature engineering yang umumnya digunakan memiliki keterbatasan, terutama dalam hal keberlakuan pada dataset tertentu saja tanpa kemampuan generalisasi yang luas. Hal ini disebabkan oleh variasi antara citra, seperti perbedaan sudut pandang, skala, kondisi pencahayaan, deformasi objek, dan faktor-faktor lainnya (Gilang Perwati, Suarna, and Suprapti 2024).

Convolutional Neural Network (CNN) untuk sistem klasifikasi tanaman anggrek. CNN dapat ditempatkan sebagai satu dari banyak metode *Deep Learning* yang ada. Teknologi *Deep Learning* bisa diterapkan untuk menemukan dan mengidentifikasi objek dalam gambar digital. *Deep Learning* ialah salah satu cabang dari disiplin ilmu *Machine Learning*. Simple Nya, *Deep Learning* adalah menerapkan konsep dasar *Machine Learning* dengan menggunakan algoritma jaringan syaraf tiruan yang memiliki lebih banyak lapisan. Jumlah lapisan tersembunyi yang digunakan antara lapisan *input* dan lapisan output membuat jaringan ini dapat disebut sebagai jaringan *neural* yang dalam. Dalam beberapa tahun terakhir, penggunaan *Deep Learning* telah mengungkapkan kemampuan yang mengesankan. Hal ini terutama disebabkan oleh komputasi yang lebih kuat, kumpulan data yang besar, dan metode untuk melatih jaringan yang lebih (Kolluru et al., 2018).

Oleh karena itu, penggunaan *Convolutional Neural Network* (CNN) diharapkan dapat membantu mempermudah dalam pengklasifikasian jenis-jenis tanaman anggrek di Anggrek Kuningan. Hal ini tidak hanya mendukung pengelola Anggrek Kuningan untuk mengetahui jenis tanaman anggrek nya namun hal ini juga berdampak kepada pelanggan yang kurangnya pengetahuan terhadap jenis-jenis tanaman anggrek.

Berdasarkan pemaparan latar belakang diatas, maka penulis mengangkat judul penelitian ini adalah **"Implementasi Algoritma *Convolutional Neural Network* (CNN) Pada Sistem Klasifikasi Bunga Anggrek"**.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang diatas maka penulis dapat mengidentifikasi masalah yang didapat yaitu:

1. Belum adanya aplikasi pendeteksi tanaman anggrek menyebabkan pengelola masih harus melakukan pengecekan secara manual melalui pencarian di *Google* untuk membedakan jenis tanaman anggrek.
2. Pengunjung mengalami kesulitan dalam membedakan jenis bunga anggrek secara manual karena kemiripan bentuk dan warna.

1.3 Rumusan Masalah

Berdasarkan latar belakang diatas, maka penulis dapat merumuskan masalah yang didapat yaitu:

1. Bagaimana membangun aplikasi untuk mendeteksi dan mengklasifikasikan jenis tanaman anggrek?
2. Bagaimana mengimplementasikan algoritma Deep Learning menggunakan *Convolutional Neural Network* (CNN) untuk mendeteksi objek tanaman anggrek?

1.4 Batasan Masalah

Dalam pembahasan dan permasalahan yang terjadi, penulis menetapkan beberapa Batasan masalah seperti dibawah ini :

1. Algoritma yang digunakan *Convolutional Neural Network* (CNN) untuk pengklasifikasian jenis tanaman.
2. Arsitektur CNN yang digunakan yaitu *MobileNetV2*.
3. Data yang digunakan dalam penelitian ini adalah 3 tanaman anggrek *Phalaenopsis Amabilis*, *Vanda* dan *Dendrobium*.
4. Dataset yang digunakan yang digunakan 600.
5. Data *training* 70%, data *testing* 15% data *validation* 15%.
6. Dataset *primary* diambil secara langsung di tempat penelitian digunakan sebanyak 20%.
7. Dataset *secondary* diambil dari Kaggle sebanyak 30% dan roboflow sebanyak 50%.
8. Pelabelan gambar yang digunakan yaitu annotations.csv.

9. Aplikasi yang di buat berbasis android dengan versi android minimal 5.0/5.1.

1.5 Tujuan Penelitian

Berdasarkan masalah yang telah dirumuskan diatas maka tujuan penelitian ini adalah:

1. Membangun aplikasi berbasis Android yang mampu mendeteksi dan mengklasifikasikan jenis tanaman anggrek.
2. Mengimplementasikan algoritma *Deep Learning* menggunakan *Convolutional Neural Network* (CNN) untuk mendeteksi objek tanaman anggrek.

1.6 Manfaat Penelitian

1.6.1 Manfaat Teoritis

Manfaat teoritis dari pembuatan skripsi ini diharapkan mampu menjadi acuan terhadap penelitian sejenis.

1. Sebagai suatu tahapan proses belajar pada suatu masalah yang diteliti di dunia nyata.
2. Mengimplementasikan ilmu yang telah diperoleh selama menempuh Pendidikan di Fakultas Ilmu Komputer Universitas Kuningan dengan membuat penelitian secara ilmiah dan sistematis.
3. Memahami lebih dalam mengenai metode *algoritma Convolutional neural network* untuk klasifikasi citra.

1.6.2 Manfaat Praktis

Manfaat praktis dari penelitian ini terhadap toko anggrek Kuningan diantaranya:

1. Memberikan pelayanan yang lebih cepat kepada pelanggan dalam mengetahui jenis-jenis tanam anggrek.
2. Memudahkan pengelola dalam mengidentifikasi jenis-jenis tanaman anggrek.

1.7 Pertanyaan Penelitian

Adapun pernyataan penelitian yan ditanyakan dalam penelitian skripsi ini adalah sebagai berikut:

1. Apakah algoritma *Convolutional Neural Network* (CNN) dapat mengklasifikasikan jenis bunga anggrek dengan tingkat akurasi yang tinggi?
2. Apakah penerapan CNN dapat meningkatkan proses klasifikasi jenis bunga anggrek dibandingkan metode manual atau tradisional?

1.8 Hipotesis Penelitian

Hipotesis merupakan jawaban sementara terhadap rumusan masalah penelitian, hipotesis pada penelitian ini yaitu sebagai berikut:

1. Algoritma CNN mampu mengklasifikasikan jenis bunga anggrek dengan tingkat akurasi yang tinggi.
2. Penerapan CNN meningkatkan proses klasifikasi jenis bunga anggrek dibandingkan metode manual atau tradisional.

1.9 Metodologi Penelitian

Metodologi penelitian adalah proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk keperluan penelitian. Adapun metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1.9.1 Metode Pengumpulan Data

Teknik yang dilakukan untuk mengumpulkan data, yaitu:

1. Metode Observasi

   Metode yang dipilih untuk mengumpulkan data ialah melakukan metode observasi dengan datang langsung ke Anggrek Kuningan untuk mengamati dan mendapatkan informasi seputar masalah yang akan diambil. Sehingga memperoleh suatu data jenis-jenis tanaman anggrek yang cukup relevan dan akurat.

2. Metode Wawancara

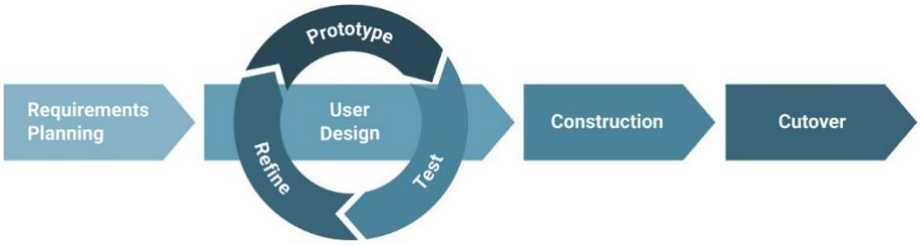
Metode pengumpulan data dengan melakukan tanya jawab kepada Bapak Ujang Moch Furkon dari Anggrek Kuningan selaku pemilik usaha tanaman hias guna mendapatkan informasi seputar tanaman anggrek. Sekaligus dengan informasi seputar tanaman anggrek yang ada di Anggrek Kuningan.

3. Metode Studi Pustaka

Studi pustaka dilakukan untuk menyusun laporan penelitian dengan melakukan pengumpulan data dan berupa pengumpulan informasi yang relevan dan di peroleh dari beberapa sumber seperti buku, jurnal dan internet yang berhubungan dengan kebutuhan penelitian dengan tujuan untuk melengkapi data-data dan informasi yang di butuhkan dalam penelitian.

1.9.2 Metode Pengembangan Sistem

Metode pengembangan system yang digunakan oleh penulis dalam penelitian ini adalah dengan metode *Rapid Application Devlovment* (RAD). Metode RAD merupakan proses pengembangan perangkat lunak yang bersifat incremental terutama untuk waktu pengerjaan yang pendek. Tahapan pada metode RAD terdiri dari *Requirement Planning, User Design, Construction, dan Cutover*. Gambar dibawah menunjukan secara keseluruhan arsitektur yang dimiliki RAD (Natbais and Umbu 2023).



Gambar 1.1 Arsitektur Rapid Application Development (Atmaja, Faizah, and Kambry 2023)

Berdasarkan gambar 1.1 *metode Rapid Application Development* memiliki beberapa tahapan yaitu:

1. *Requirement Planning*

   Tahapan ini merupakan tahapan paling awal dimana penulis mengumpulkan data kebutuhan terhadap *system* yang akan di kembangkan. Pada tahapan ini dilakukan proses pengumpulan data yang akan dilakukan pada penelitian ini yaitu dengan cara wawancara, observasi dan studi pustaka.

2. *User Design*

   Tahap ini lebih difokuskan pada pembuatan desain untuk perencanaan arsitektur sistem. Peneliti akan membuat prototipe sistem informasi pada tahap *User Design*. Setelah perencanaan persyaratan selesai, *prototipe* diuji. Jika terjadi ketidaksesuaian dalam hasil pengujian *prototipe*, perbaikan dapat dilakukan.

3. *Construction*

   Tahap ketiga yaitu *construction* merupakan tahap pengembangan. Pada tahap ini dilakukan perancangan perangkat lunak berdasarkan pada prototipe yang telah dibuat sesuai dengan hasil *requirement planning*.

4. *Cutover*

   Pada tahapan ini adalah tahap terakhir yang penulis lakukan setelah rancangan sistem disetujui oleh user, dan melakukan perbaikan yang diperlukan.

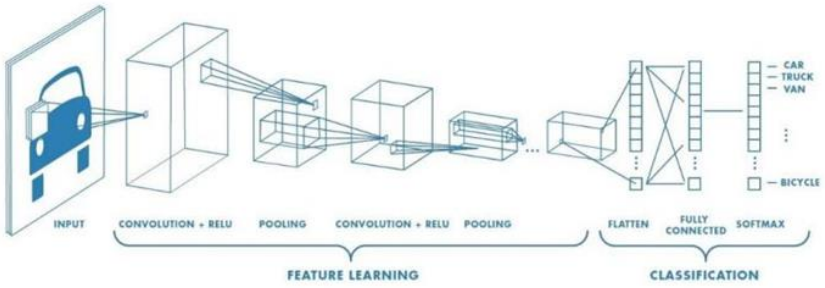
1.9.3 Metode Penyelesaian Masalah

Berdasarkan tujuan utama dari penyusunan proposal penelitian adalah membuat sebuah aplikasi yang dapat mengidentifikasi tanaman anggrek dengan menerapkan algoritma *Convolutional Neural Network* dan penggunaan *TensorFlow* sebagai *library* yang akan membantu dalam menyematkan model pada aplikasi mobile.

1. Algoritma *Convolutional Neural Network* (CNN)

   Convolutional Neural Network (CNN) terinspirasi oleh proses-proses biologi dimana pola konektivitas antar *neurons*

menyerupai organisasi visual *cortex* pada binatang. *Cortical neurons* menanggapi stimulasi hanya dalam suatu area terbatas pada bidang visual atau bidang reseptif (*receptive field*). Bidang-bidang reseptif *neurons* tumpang tindih secara parsial (*partially overlap*) sedemikian hingga mencakup seluruh bidang reseptif. Oleh karena itu, CNN banyak diaplikasikan pada analisis citra. 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*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*. Gambar 1.2 adalah jaringan arsitektur *Convolutional Neural Network* (Rizki et al. 2023).

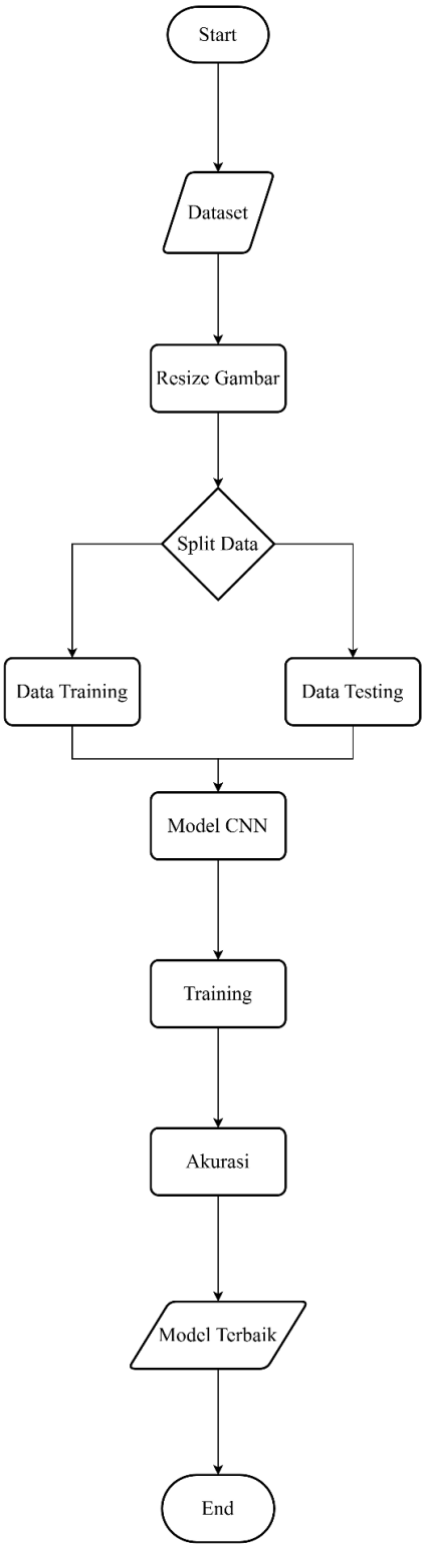


Gambar 1.2 Arsitektur Convolutional Neural Network (Riyadi, Wardhani, and Widayati 2021)

Berdasarkan gambar diatas, Tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU (*Rectifier Linear Unit*), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully*

connected neural network, dan dari *fully connected network* adalah *output class*.

Flowchart Sistem CNN adalah alur lebih detail bagaimana datasets masuk sebagai *input* pada program di olah oleh layer konvolusi, pooling, dropout, untuk di ambil *feature* dari masing-masing gambar dalam bentuk matriks, kemudian dari matriks tersebut di ubah menjadi *vector* yang kemudian digunakan untuk *input* pada jaringan syaraf tiruan. Berikut gambar 1.3 merupakan *flowchart* sistem CNN.



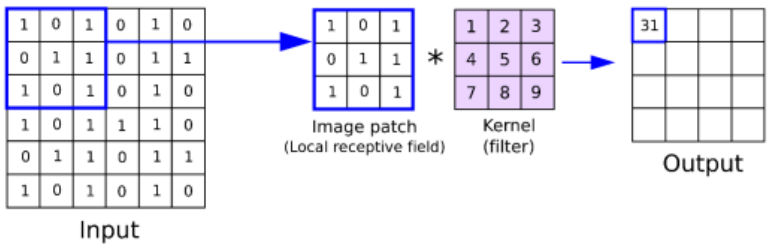
Gambar 1.3 Flowchart Convolutional Neural Network(Patel and Patel 2019)

a. *Input Layer*

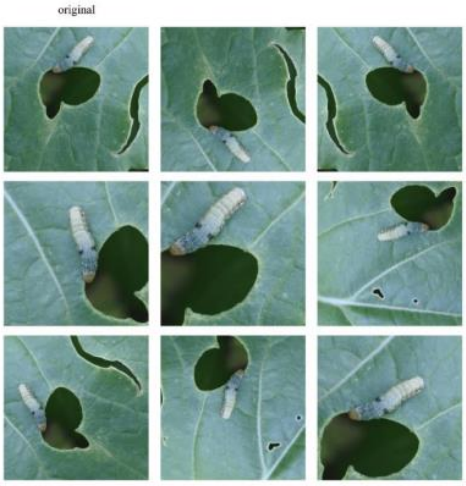
Citra dengan dimensi 128x128x4. Kemudian data citra yang telah di pra proses dimasukan ke dalam *input layer*. Nilai *input* dan *output* dari layer ini adalah 128x128 dengan nilai matriks [-1,0,1] dan *channel* RGB.

b. *Convolutional*

Convolution layer komponen utama CNN, terdiri dari berbagai filter yang diinisialisasi secara acak untuk melakukan operasi konvolusi dan ekstraksi fitur untuk mempelajari representasi fitur dari gambar yang dimasukkan. Nilai awal acak ini akan diperbarui (di*update*) selama pelatihan. Proses ini dikalikan dan bergerak dari sudut kiri atas ke kanan bawah di bagian gambar. Untuk membuat representasi fitur, setiap gambar kecil yang dihasilkan dari hasil konvolusi digunakan sebagai *input*. Ini memungkinkan CNN untuk mengidentifikasi objek di mana pun posisinya muncul pada gambar 1.4.



Gambar 1.4 Ilustrasi Layer Konvolusi



Gambar 1.5 Ilustrasi Data Argumentasi

Red

0	0	0	0	0	0
0	70	35	28	42	0
0	45	25	30	48	0
0	31	28	32	49	0
0	30	30	35	50	0
0	0	0	0	0	0

Green

0	0	0	0	0	0
0	107	59	45	58	0
0	76	42	45	65	0
0	51	41	46	65	0
0	42	41	48	64	0
0	0	0	0	0	0

Blue

0	0	0	0	0	0
0	112	69	51	59	0
0	79	48	47	63	0
0	56	43	46	60	0
0	47	42	46	59	0
0	0	0	0	0	0

Gambar 1.6 Nilai Pixel RGB Dataset [9]

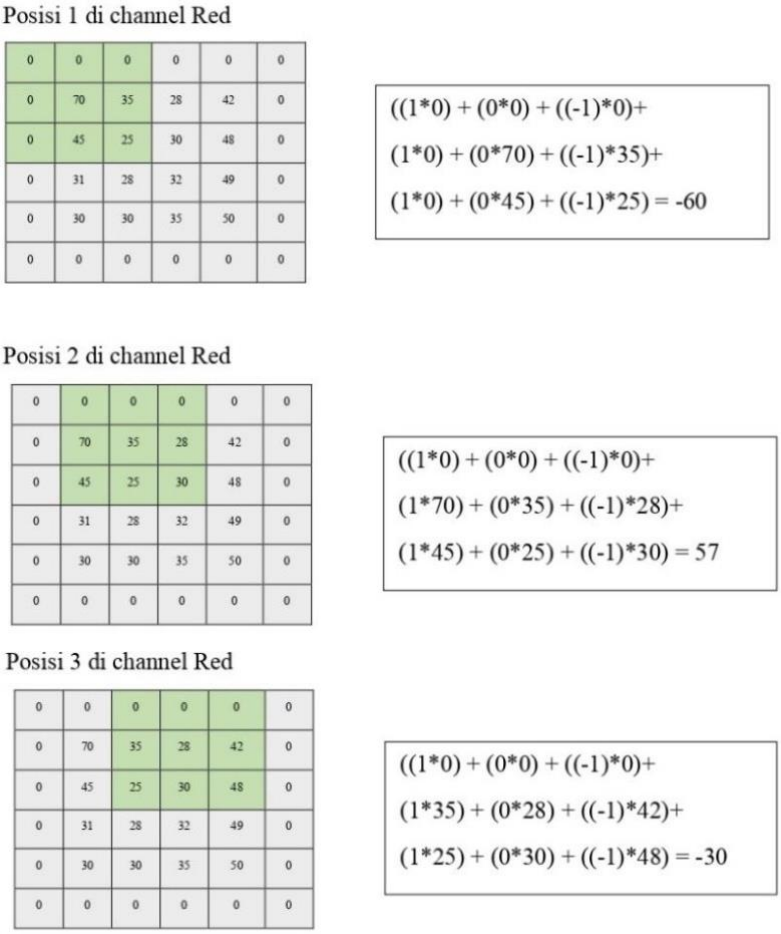
Pada percobaan ini digunakan kernel 3 x 3 dengan nilai seperti pada gambar 5 dibawah ini:

1	0	-1
1	0	-1
1	0	-1

Gambar 1.7 Kernel 3x3 [9]

Langkah selanjutnya dilakukan proses perhitungan di setiap *channel* dengan dikalikan dengan kernel ukuran 3x3 pada gambar 1.7. Tahapan ini dilakukan berulang dengan pergeseran kernel sebanyak 1 strides di setiap *channel* Nya sehingga didapatkan perhitungan di setiap *channel* dengan nilai sebagai berikut:

Channel Red



Gambar 1.8 Ilustrasi Perhitungan Layer (Konvolusi Channel Red) [9]

Gambar 1.8 menjelaskan ilustrasi tahapan konvolusi. Sebagai contoh proses perhitungan dilakukan pada posisi 1 *channel red* yang diberi tanda kotak hijau akan dikalikan dengan nilai kernel 3x3 pada gambar 6. Sehingga perkalian antara matrix

3x3 yang dikalikan sesuai dengan baris dan kolomnya mendapatkan nilai -60. Perhitungan dilanjutkan sama seperti pada posisi 1 sampai 16 sehingga didapatkan hasil seperti pada gambar di bawah ini:

Red

-60	57	-30	58
-88	56	-51	90
-83	9	-64	97
-58	-6	-41	67

Gambar 1.9 Hasil perhitungan konvolusi channel red [9]

Perhitungan yang sama dilakukan pada *channel green* dan *blue* dimana hasil perhitungannya seperti di bawah ini:

Green

-101	93	-22	90
-142	98	-46	136
-124	30	-70	139
-82	-1	-47	94

Gambar 1.10 Hasil perhitungan konvolusi channel Green [9]

Blue

-117	93	-5	98
-160	103	-22	144
-133	43	-49	139
-85	11	-34	92

Gambar 1.11 Hasil perhitungan konvolusi channel Blue [9]

Setiap *channel* yang sudah melalui tahap perhitungan pada gambar 6, 7 dan 8 akan dijumlahkan sehingga menghasilkan *output* total seperti pada gambar dibawah ini:

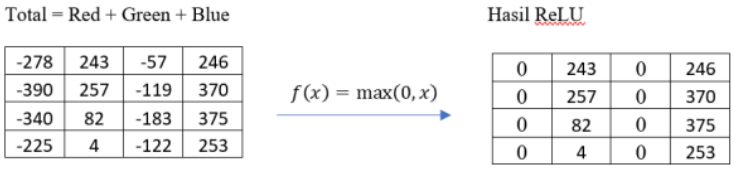
Total = Red + Green + Blue

-278	243	-57	246
-390	257	-119	370
-340	82	-183	375
-225	4	-122	253

Gambar 1.12 Total perhitungan Convolutional [9]

c. *ReLU*

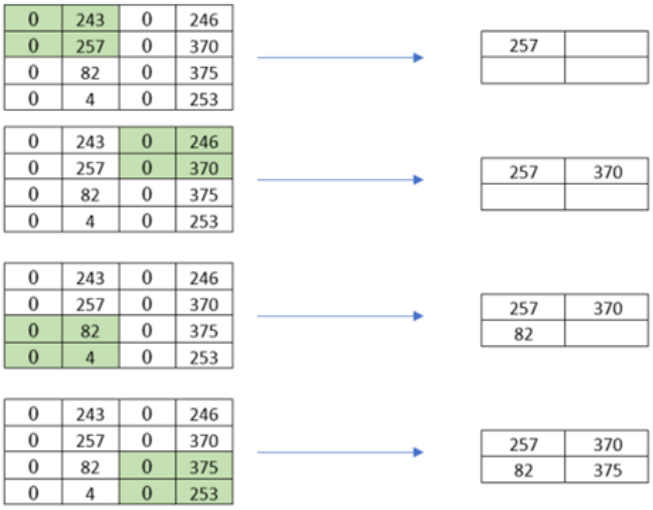
Dengan *activation ReLU* maka setiap nilai yang *negative* akan diubah menjadi angka 0 sehingga didapatkanlah hasil *convolutional* seperti dibawah ini:



Gambar 1.13 Hasil perhitungan Convolutional [9]

d. *Pooling Layer*

Setelah *ReLU*, biasanya pada layer berikutnya diterapkan *max Pooling*. Tujuan dari penggunaan *max pooling* adalah untuk melakukan *down sampling* dari representasi, mengurangi dimensi yang memungkinkan hal ini dilakukan untuk membantu penyesuaian secara berlebihan dengan menyediakan bentuk representasi dengan nilai maksimum dari *input*. Selain itu, *max pooling* mengurangi biaya komputasi dengan mengurangi jumlah parameter untuk dipelajari dan memberikan inventarisasi translasi dasar ke representasi *internal*. Dari *output ReLU* akan dijadikan sebagai *input* pada *pooling layer*, pada tahapan ini menggunakan *max pooling* dengan kernel *size* 2x2 dengan *stride* 2 dengan *output* dari tahapan konvolusi 4x4x32. Ilustrasi proses *max pooling* ditujukan seperti gambar dibawah:



Gambar 1.14 Ilustrasi Perhitungan Layer (Max pooling)

e. *Dropout Layer*

Dropout berfungsi untuk mengurangi kompleksitas model yang telah dibangun dengan membuang *neural network* yang tidak terpakai. Ukuran model akan tetap 2x2x32, tidak ada perubahan.

f. *Flatten Layer*

Mengubah nilai masukan menjadi sebuah array hasil *pooling*, setiap hasil dari proses *pooling layer* akan diubah menjadi bentuk array satu dimensi.



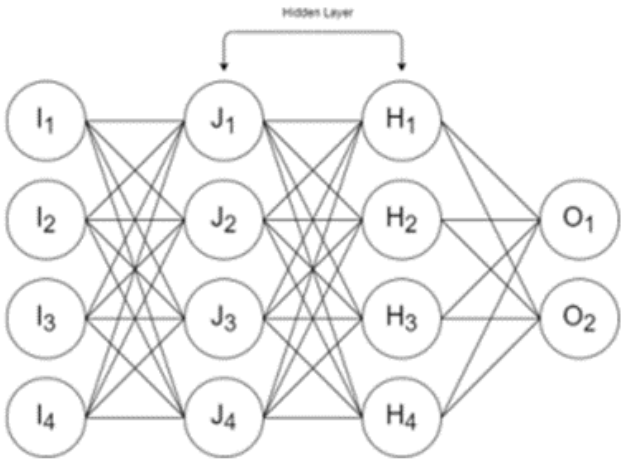
Gambar 1.15 Ilustrasi Proses Flatten [9]

g. *Full Connected Layer*

Fully connected layer adalah merupakan *multi-layer perceptron* tradisional yang menggunakan fungsi aktivasi *SoftMax* pada lapisan *output*. Istilah "*Fully Connected*" menyatakan bahwa setiap *neuron* pada *layer* sebelumnya

adalah terhubung ke dalam setiap *neuron-neuron* pada lapisan berikutnya.

Output dari *convolution* dan pooling *layer* merepresentasikan *high level features* hasil ekstraksi dari input gambar. *Layer* ini pada dasarnya mendapatkan *volume input* (*convolution*, *ReLU* atau *pooling Layer*) dan menghasilkan vektor dimensi N dimana N adalah jumlah kelas. Tujuannya adalah menggunakan fitur-fitur tersebut untuk mengklasifikasikan gambar ke dalam berbagai kelas berdasarkan yang sudah diinisialisasi pada data *training*. Berikut ilustrasi *fully connected* layer seperti gambar dibawah:



Gambar 1.16 Ilustrasi proses dense + SoftMax

Tahapan akhir model CNN yang dibangun ini adalah *Dense* dengan fungsi aktivasi *SoftMax*. Berikut perhitungan dari setiap *hidden layer*.

$$\sum_{i=1}^{N} I_i * v_{ij} = J_i$$

$$\sum_{i=1}^{N} J_i * w_{ij} = H_i$$

$$\sum_{i=1}^{N} H_i * x_{ij} = O_i$$

Gambar 1.17 Ilustrasi proses dense + SoftMax [9]

11, 12, 13 ,14 adalah nilai *output* pada proses *flatten* yang kemudian dilakukan *fully connected layer* dengan perhitungan setiap *hidden layer* sebagai berikut:

J_1 = (257*0,1) + (370*0,1) + (82*0,1) +(375*0,1) = 108,4

J_2 = (257*0,4) + (370*0,4) + (82*0,4) +(375*0,4) = 433,6

J_3 = (257*0,5+ (370*0,5) + (82*0,5) +(375*0,5) = 524

J_4 = (257*0,3) + (370*0,3) + (82*0,3) +(375*0,3) = 325,2

Hidden layer pertama di ilustrasikan dengan j1, j2, j3, j4 dimana setiap neuron dari 11,12,13 dan 14 dikalikan dengan nilai *weight* yang berbeda-beda sehingga menghasilkan nilai j1=108,4, j2=433,6, j3=542, dan j4=325,2. Yang setiap *neuron* dari j1, j2, j3, dan j4 ini akan dikalikan lagi dengan *weight* yang berbeda-beda untuk menghasilkan nilai h1, h2, h3, h4 sebagai *hidden layer* selanjutnya.

H_1 = (108,4*0,3) + (433,6*0,3) + (542*0,3) + (325,2*0,3) = 422,76

H_2 = (108,4*0,5) + (433,6*0,5) + (542*0,5) + (325,2*0,5) = 704,06

H_3 = (108,4*0,2) + (433,6*0,2) + (542*0,2) + (325,2*0,2) = 281,84

H_4 = (108,4*0,4) + (433,6*0,4) + (542*0,4) + (325,2*0,4) = 563,68

Yang setiap *neuron* dari h1, h2, h3, dan h4 ini akan dikalikan lagi dengan *weight* yang berbeda-beda untuk menghasilkan nilai o1, dan o2 sebagai hidden layer selanjutnya.

O_1 = (422,76*0,1) + (704,06*0,1) + (281,84*0,1) + (563,68*0,1) = 1973,28

O_2 = (422,76*0,3) + (704,06*0,3) + (281,84*0,3) + (563,68*0,3) = 591,864

langkah selanjutnya adalah perhitungan *SoftMax* dengan rumus eksponensial o1 dibagi dengan total eksponensial o1 dan eksponensial o2. Tahapan ini dilakukan 2 kali baik pada o1 dan o2 dengan rumus dan ilustrasi sebagai berikut:

$$s(o_i) = \frac{e^{0_i}}{\sum_{j=1}^{n} e^{o_j}}$$

$$s(o_i) = \frac{e^{0_i}}{\sum_{j=1}^{n} e^{o_j}} \qquad s(o_1) = \frac{e^{1973,28}}{e^{1973,28} + e^{591,864}} \qquad s(o1) = 0{,}8$$

$$s(o_2) = \frac{e^{0_i}}{\sum_{j=1}^{n} e^{o_j}} \qquad s(o_2) = \frac{e^{591,864}}{e^{1973,28} + e^{591,864}} \qquad s(o2) = 0{,}2$$

$$s(o1) + s(o2) = 1$$

$$0.8 + 0.2 = 1$$

Sehingga didapatkanlah bobot nilai probabilitas yang lebih besar pada o1 yaitu 0.8 yang berarti input citra yang dimasukan diprediksi adalah *with pest*.

h. Confusion Matrix

Penentuan performa suatu model klasifikasi baik atau tidaknya dapat dilihat dari parameter pengukuran performa nya berdasarkan tingkat akurasi, *recall*, dan presisi. *confusion* matriks adalah cara untuk menghitung faktor-faktor tersebut dengan sebuah matriks. Dalam pengukuran biasanya digunakan salah satu caranya yaitu dengan *Confusion-matrix* pada gambar 5.

		Kejadian Sebenarnya	
		P	N
Hipotesis Kejadian	P	True Positive	False Positive
	N	False Negative	True Negative

Gambar 1.18 Confusion Matrix [9]

Berdasarkan gambar 5 di atas terdapat beberapa nilai di dalam matriks yaitu (TP) "*True Positive*", (TN) "*True Negative*", (FP) "*False Positive*", dan (FN) "*False Negative*", semua

kemungkinan kejadian sebenarnya *positif* (P) dan semua kemungkinan kejadian sebenarnya *negatif* (N). Nilai tersebut dapat digunakan untuk menghitung akurasi dengan persamaan dibawah

$$Asukrasi = \frac{TP + TN}{TP + TN + FP + FN} x\ 100\%$$

Akurasi dijadikan patokan sebagai penilaian sebagaimana akurat suatu model melakukan klasifikasi. Sementara untuk menghitung level presisi prediksi kejadian dapat digunakan persamaan dibawah:

$$Presisi = \frac{TP}{TP + FP} x\ 100\%$$

Presisi menunjukkan seberapa tepat suatu model memprediksi kejadian *positif* dalam serangkaian kegiatan prediksi. Perhitungan presisi biasanya digunakan untuk pengembangan model prediksi hujan di suatu daerah. Selain presisi dan akurasi, untuk dapat melihat lebih detail lagi kinerja suatu sistem, *recall* atau *sensitivity* sistem terhadap suatu kelas juga dapat dilihat. *Recall* dapat dihitung dengan menggunakan persamaan.

$$Sensitifitas\ Prediski = \frac{TP}{TP + FN} x\ 100\%$$

F1-Score adalah *harmonic mean* dari *precision* dan *recall*. Yang secara matematik dapat ditulis seperti persamaan berikut

$$F1\ Score = 2x \frac{presisi\ x\ recall}{presisi + recall} x\ 100$$

Nilai terbaik F1-Score adalah 1.0 dan nilai terburuk nya adalah 0. Secara representasi, jika F1-Score punya skor yang

baik mengindikasikan bahwa model klasifikasi kita punya *precision* dan *recall* yang baik.

1.10 Jadwal Penelitian

Jadwal pelaksanaan penelitian dibuat dengan tahapan yang jelas dalam bentuk bar chart

Tabel 1.1 Jadwal Kegiatan

	Desember 2023				Januari 2024				Februari 2024				April 2025				Mei 2025			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Requirement Planning																				
- Wawancara																				
- Observasi																				
- Studi Pustaka																				
User Design																				
- Prototype																				
- Uji Ptototype																				
Construction																				
- Perancangan																				
- Implementasi Sistem																				
Cutover																				
- Pengujian Sistem																				
- Pelatihan User																				
Seminar Hasil																				
Sidang																				

1.11 Sistematika Penelitian

BAB I : PENDAHULUAN

Bab Pendahuluan menguraikan mengenai Latar Belakang, Identifikasi Masalah, Rumusan Masalah, Batasan Masalah, Tujuan dan Manfaat Penelitian, Pertanyaan Penelitian, Hipotesis Penelitian, Metode Penelitian, Jadwal Penelitian dan Sistematika

BAB II : LANDASAN TEORITIS

Bab Landasan Teori ini, menjelaskan tentang teori – teori yang melandasi penulisan skripsi ini.

BAB III : ANALISA DAN PERANCANGAN

Bab Analisis dan Perancangan, Menguraikan mengenai proses perancangan yang dilakukan. Adapun yang dibahas pada bab ini mencakup perancangan sistem, program berbasis mobile dan perancangan pada aplikasi ujian *online* penentuan level kursus.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab Implementasi dan Pengujian berisi implementasi perancangan sistem dari hasil analisis dan perancangan yang sudah dibuat, serta menguji sistem untuk menemukan kelebihan dan kekurangan pada sistem yang dibuat.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi simpulan terhadap aplikasi yang dibuat secara keseluruhan, dan dikemukakan saran – saran untuk perbaikan serta pengembangan aplikasi.