

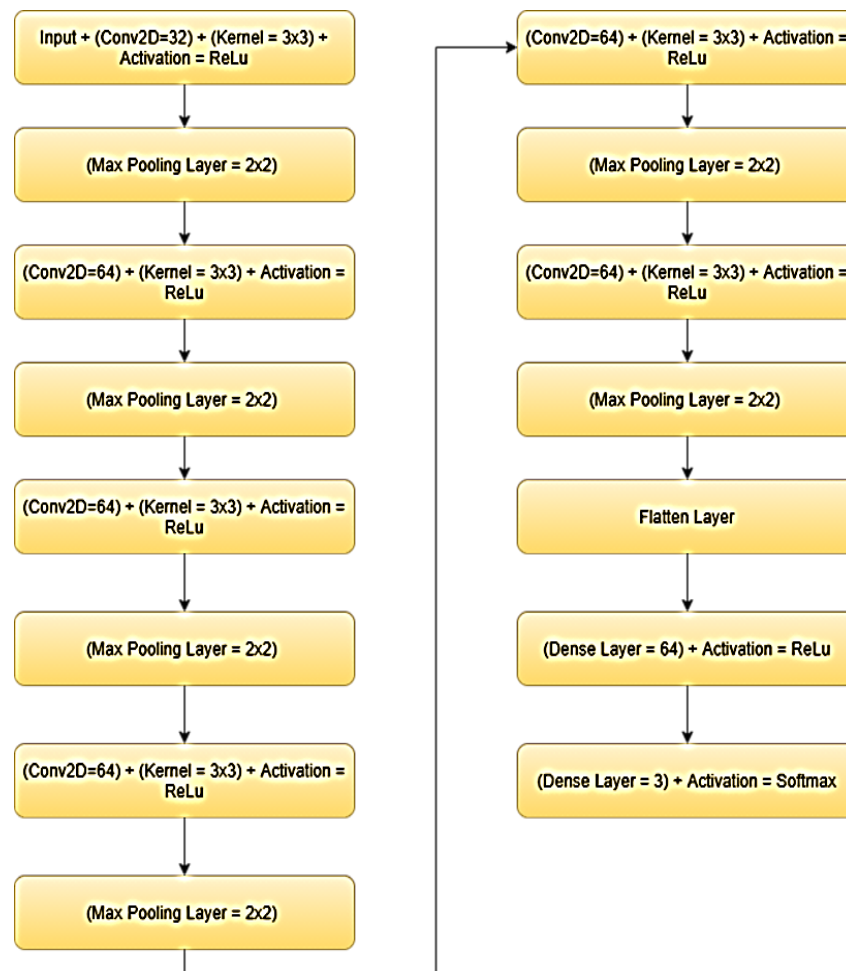
BAB III

ANALISA PERANANGAN SISTEM

3.1 Perancangan Model

3.1.1 Rancangan Yang Diajukan

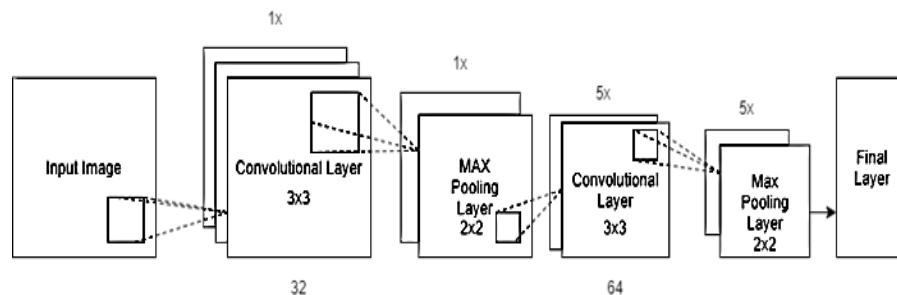
Rancangan yang dibuat memiliki 6 *konvolusional layer* dengan 2 jumlah filter yang berbeda. Selanjutnya *output* dari ke 6 *layer* tersebut akan dijadikan input di *layer flatten* untuk mengubahnya menjadi bentuk vektor. Setelah itu vektor tersebut akan dijadikan *input dilayer dense* dan mengaktifkannya dengan aktivasi dari *ReLU*. Lalu disusul dengan *layer dense* ke 2 yang menggunakan aktivasi *softmax* untuk menghitung nilai probabilitas dari setiap kelas. Pada gambar berikut ini merupakan Rancangan yang diajukan.



Gambar 3.1 Rancangan yang diajukan

3.1.2 Fitur Ekstrasi model

Sesuai gambar 3.2 fitur ekstrasi model menggunakan 1 konvolusi *layer* dengan *kernel* 3×3 per 32 untuk mengkonvolusi input gambar lalu diambil nilai maksimalnya dengan metode *max pooling* dengan *kernel* 2×2 lalu dilakukan konvolusi dengan ukuran 3×3 per 64 dan *max pooling* dengan ukuran 2×2 sebanyak 5x lalu hasil dari proses tersebut akan menjadi input bagi *final layer* (*Flatten dan Dense layer*).



Gambar 3.2 Feature ekstrasi model

Untuk Flatten sendiri memiliki 1 layer yang memiliki fungsi mengkonversi hasil dari konvolusi dan *pooling* ke *vector* kemudian vektor tersebut akan menjadi *input* untuk *dense* yang berjumlah 2 layer.

Tabel 3.1 Arsitektur feature yang diajukan

Layer		Fitur dan ukuran	Ukuran Kernel	Aktivasi
Input	Image	$256 \times 256 \times 3$	-	-
1	Convolutional	$254 \times 254 \times 32$	3×3	ReLU
	Max Pooling	$127 \times 127 \times 32$	2×2	ReLU
2	Convolutional	$125 \times 125 \times 64$	3×3	ReLU
	Max Pooling	$62 \times 62 \times 64$	2×2	ReLU
3	Convolutional	$60 \times 60 \times 64$	3×3	ReLU
	Max Pooling	$30 \times 30 \times 64$	2×2	ReLU
4	Convolutional	$28 \times 28 \times 64$	3×3	ReLU
	Max Pooling	$14 \times 14 \times 64$	2×2	ReLU
5	Convolutional	$12 \times 12 \times 64$	3×3	ReLU
	Max Pooling	$6 \times 6 \times 64$	2×2	ReLU

Tabel 3.2 Arsitektur feature yang diajukan (lanjutan)

6	<i>Convolutional</i>	$4 \times 4 \times 64$	3×3	<i>ReLU</i>
	<i>Max Pooling</i>	$2 \times 2 \times 64$	2×2	<i>ReLU</i>
7	<i>Flatten</i>	256	-	-
8	<i>Dense 1</i>	64	-	<i>ReLU</i>
9	<i>Dense 2</i>	3	-	<i>Softmax</i>

3.2 Jenis, Sifat dan Pendekatan Penelitian

Tahap studi pustaka yaitu dengan pengumpulan referensi pustaka dan pemahaman yang berkaitan dengan metode CNN dan jurnal-jurnal penelitian terkait serta pedoman klasifikasi, pengolahan citra, dan lain – lain.

1. Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan dengan mempelajari data Dataset Bawang Merah yang diunggah oleh Jumrayanti di *website kaggle* menjadi pertimbangan dalam melakukan klasifikasi data. Data ini akan menjadi variabel input dalam penelitian.

2. Tahap perancangan sistem

Dalam tahap ini dilakukan analisis data pada daun tanaman kentang dan membuat perancangan dengan *flowchart* dari model yang akan dibuat.

3. Tahap Implementasi

Dalam tahap ini dilakukan implementasi menggunakan metode CNN dengan bahasa pemrograman *python* dan ditampilkan dengan *ui* dari *react-js*. Implementasi ini disesuaikan dengan konsep dalam tahapan analisis dan perancangan serta pemodelan data.

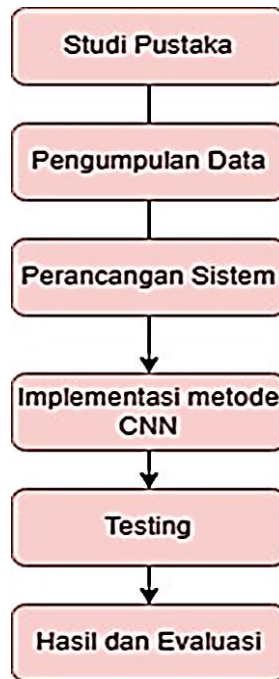
4. Tahap testing/pengujian

Setelah berhasil melakukan implementasi maka akan dilakukan pengujian pada sistem untuk mengetahui apakah sistem dapat digunakan sebagai mana mestinya.

5. Tahap Hasil dan Evaluasi

Dalam tahap terakhir hasil yang diberikan sistem akan dievaluasi dan dipelajari kemudian dapat dibuat sebagai bahan referensi untuk penelitian selanjutnya.

Pada gambar dibawah ini merupakan Tahap dari penelitian ini:



Gambar 3.3 Metode penelitian

3.3 Kebutuhan Fungsional

Kebutuhan Fungsional atau kebutuhan yang diperlukan untuk tahap perancangan model adalah sebagai berikut :

1. Perancangan Model menggunakan aplikasi *Draw.io* atau dapat menggunakan *Power Designer* maupun *Microsoft Visio*.
2. *Anaconda 3* sebagai Platform penyedia bahasa *Python*.
3. *Text Editor* menggunakan *Visual Studio Code* dan *Jupyter Notebook*.
4. Web Browser untuk menampilkan Model seperti *Chrome* atau *Mozilla Firefox*.
5. *Pycharm* sebagai *Backend* atau API yang akan dijalankan dari sistem yang akan dibuat.

3.4 Kebutuhan Non Fungsional



Kebutuhan Non Fungsional atau kebutuhan perangkat keras yang digunakan penulis untuk melakukan penelitian adalah sebagai berikut:

1. *Laptop Asus X43U-VX144D Series.*
2. *Prosesor Intel E-Series E-350 dengan AMD Brazos dual core c50 Graphics 6250 1 Hz.*
3. *Ram 2 GB*
4. *HDD 320GB.*

3.5 Pengumpulan data

Data yang diambil berasal dari *website kaggle* bernama Dataset Bawang Merah yang diunggah oleh Jumrayanti pada tahun 2022. di Dalam Dataset memiliki jumlah 208 data gambar terbagi dengan 2 kelas/kondisi yaitu, Bercak Unggu (*Purple Spot*), dan Layu Fusarium (*Fusarium wilt*). Data tersebut nantinya akan diinput kemudian di proses dengan model *Data Augmentation* dan Metode CNN untuk mengklasifikasi sekaligus mengidentifikasi Data – Data tersebut. Berikut adalah tabel gambar dari 2 kondisi daun tanaman bawang merah tersebut :

Tabel 3.3 Dataset

No.	Gambar Data	Kondisi/Kelas	Jumlah Data
1.		Bercak Unggu (<i>Purple Spot</i>)	104 Gambar
2.		Layu Fusarium (<i>Fusarium wilt</i>).	104 Gambar
Total		2 Kelas/Kondisi	208 Gambar

3.6 Data Augmentation

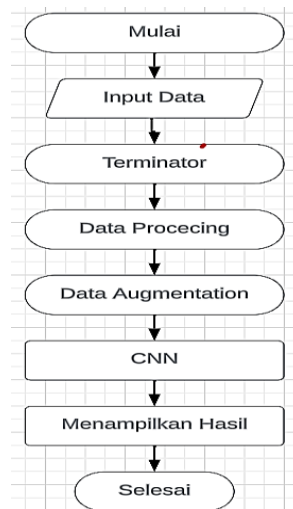
Augmentasi data adalah strategi yang memungkinkan praktisi untuk secara signifikan meningkatkan keragaman data yang tersedia untuk model pelatihan, tanpa benar-benar mengumpulkan data baru. Teknik augmentasi data seperti *cropping*, *padding*, dan *flipping* umumnya digunakan untuk melatih jaringan neural besar. Namun, sebagian besar pendekatan yang digunakan dalam pelatihan jaringan *neural* hanya menggunakan tipe augmentasi dasar. Sementara arsitektur jaringan neural telah diselidiki secara mendalam. Berikut contoh gambar dari data augmentasi :



Gambar 3.4 Data augmentation

3.7 Perancangan system

Data yang di input ke model akan melalui proses *data cleaning* dan *prepropecing* dengan metode data *augmentation* setelah itu data akan diproses dengan metode dan model *Convolutional Neural Network (CNN)* dengan *Library TensorFlow* dan *deep learning* setelah itu data akan ditampilkan melewati ui dari *react-js*. Berikut merupakan Gambar *flowchart* dari model yang akan dibuat :

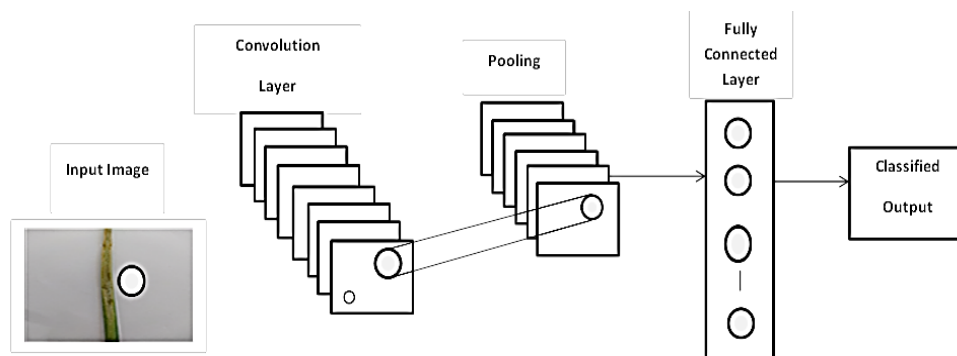


Gambar 3.4 Flowchart

Gambar diatas adalah *flowcart* yang akan dibuat.

3.8 Arsitektur CNN

Arsitektur Convolutional Neural Network yang akan dibangun untuk klasifikasi penyakit pada daun tanaman bawang merah ini terdiri dari 3 jenis *layer*, Yaitu *Convolutional Layer*, *Pooling Layer*, dan *Fully Connected Layer*. Berikut adalah gambar dari Arsitektur CNN.



Gambar 3.5 Data augmentation

Ketika menggunakan CNN, ada 4 *hyperparameter* penting yang perlu kita tentukan, yakni:

1. Ukuran kernel/filter, yaitu panjang dan lebar filter yang akan digunakan. Misalnya 3x3.

2. Jumlah filter, yaitu seberapa banyak filter yang akan kita gunakan.
3. Stride, yaitu seberapa jauh step dari filter ketika digeser.
4. *Padding*