

# DETEKSI KONDISI TANAMAN SELADA BERDASARKAN CITRA MENGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK (CNN)*

## *LETTUCE PLANT CONDITION DETECTION BASED ON IMAGE USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD*

Fachrie Reza Megantara<sup>1</sup>, Dr. Yudha Purwanto, S.T., M.T.<sup>2</sup>, Casi Setianingsih, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>fahrimegantara@telkomuniversity.ac.id, <sup>2</sup>omyudha@telkomuniversity.co.id,

<sup>3</sup>setiacasie@telkomuniversity.ac.id

### Abstrak

Metode dalam bercocok tanam mulai bervariasi pada masa kini, salah satunya dengan menggunakan metode Hidroponik. Jenis tanaman yang dapat diterapkan pada metode Hidroponik cukup banyak, salah satunya adalah tanaman Selada. Petani atau masyarakat yang menanam tanaman Selada menginginkan hasil yang baik tentunya. Perawatan dan pengelolaan tanaman yang baik menjadi faktor utama untuk menghasilkan hasil panen yang baik atau tidak. Dengan memantau tanaman satu per satu dibutuhkan waktu dan tenaga yang cukup banyak. Hal ini cukup membuat petani atau masyarakat sekalipun cukup kesulitan. Atas permasalahan tersebut, dirancang sebuah sistem yang dapat mendeteksi dari kondisi tanaman Selada tersebut dengan memanfaatkan kamera sebagai alat untuk pengambilan gambar.

Penelitian ini bertujuan untuk membuat sistem yang dapat mendeteksi dari kondisi tanaman Selada dengan menggunakan metode klasifikasi Convolutional Neural Network (CNN). Kondisi dari tanaman Selada dibedakan menjadi dua kelas kondisi yang ditandai dengan warna daun, apabila terdapat warna kuning hingga cokelat maka akan dikategorikan kelas kondisi tidak baik sedangkan daun dengan kondisi hijau segar maka akan dikategorikan kelas kondisi baik.

Hasil dari penelitian ini adalah sistem yang dapat mendeteksi kelas kondisi daun tanaman Selada dengan kondisi tidak baik dan kondisi baik. Keluaran dari sistem berupa gambar serta hasil kelas dari gambar yang telah di-input. Dengan menggunakan parameter berupa partisi data latih sebesar 90% dan data uji sebesar 10% dengan mendapatkan akurasi 99,2%.

Kata kunci : Hidroponik, Selada, *Convolutional Neural Network (CNN)*.

### Abstract

*Farming methods are starting to be varied today, one of the methods is the Hydroponic method. There are a lot of plants that can be applied to the Hydroponic method, one of the plants is Lettuce. Lettuce farmers certainly want a good Lettuce yields result. Good management and treatment for the Lettuce are the main factors for producing good or poor Lettuce yields. Because of monitoring the Lettuce one by one takes a lot of time and effort, farmers will be difficult to do that. For this problem, a system was designed that could detect the condition of the Lettuce by using a camera as a tool for taking images.*

*The purpose of this research is to create a system that can detect the condition of Lettuce plants using Convolutional Neural Network (CNN) classification method. The conditions of the Lettuce plants are divided into two class conditions which are marked by the color of the leaves, if there is yellow to brown color it will be categorized as a bad condition class while the leaves with a fresh green condition will be categorized as a good condition class.*

*The result of this research is a system that can detect the class condition of Lettuce leaves with bad conditions and good conditions. The output from the system is in the form of images and class results from the images that have been inputted. By using parameters in the form of partition training data by 90% and test data by 10% with an accuracy of 99.2%.*

*Keyword: Hydroponics, Lettuce, Convolutional Neural Network (CNN).*

### 1. Pendahuluan

Pada zaman sekarang, bertani dengan menggunakan teknik Hidroponik mulai dikembangkan dengan memanfaatkan Industri 4.0. Yang berarti, sudah memanfaatkan teknologi *Internet of Things (IoT)* dimana semua perangkat terhubung ke dalam jaringan yang dapat di kontrol dengan mudah dimapun dan kapanpun, menggunakan perangkat pengguna itu sendiri. Pentingnya pemanfaatan Industri 4.0 pada zaman sekarang sangatlah menguntungkan.

Pada penelitian sebelumnya, telah dibuat sebuah perangkat yang digunakan untuk mendeteksi pertumbuhan tanaman. Perangkat ini mengandalkan *microcontroller* dan menggunakan *interface* pada Android serta dilengkapi dengan komponen I/O lainnya [1]. Kekurangan pada penelitian tersebut adalah tidak adanya sistem *vision* pada perangkat tersebut. Maka penulis akan membuat sebuah rancangan sistem *vision* untuk mengklasifikasi kondisi dari tanaman tersebut. Dengan menggunakan algoritma *Convolutional Neural Network* (CNN), dimana akan mengklasifikasikan gambar yang telah diambil oleh kamera. Nantinya, pengguna dapat memantau kesehatan tanaman melalui sistem *monitoring* tanaman tersebut. Hal ini akan mempermudah petani atau masyarakat lain dalam pemantauan kondisi dari tanaman tersebut.

## 2. Dasar Teori

### 2.1 Hidroponik

Hidroponik merupakan sebuah budidaya menanam dengan memanfaatkan air dengan tanpa menggunakan tanah sedikit pun, sehingga nutrisi untuk tanaman tersebut didapat dari cairan khusus yang dapat memenuhi kebutuhan tanaman itu sendiri. Hidroponik juga dapat diterapkan ditengah perkotaan, karena penggunaan air yang lebih efisien karena memiliki pasokan air yang terbatas. Hidroponik tercipta menjadi sebuah inovasi budidaya pertanian, akibat terbatasnya sumber daya lahan yang tersedia, karena penanaman tanaman dengan menggunakan teknik hidroponik sangat praktis dengan tidak memanfaatkan lahan yang terlalu luas[2].

### 2.2 Tanaman Selada

Selada berasal dari negara Mesir, di wilayah Mediterania, Timur Tengah dan Asia Barat Daya. Dalam upaya untuk memperdalam genom Selada, dibangun perpustakaan kromosom buatan bakteri yang mengandung *fragmen genomic* EcoR1 dan HindIII yang besar. Area utama produksi serta konsumsi Selada adalah negara Amerika Serikat dan Eropa. Di Amerika Serikat, lebih dari  $3.18 \times 10^9$ t dipanen setiap tahunnya dan dengan kelebihan hasil panen sebesar 70% dari produksi yang berlokasi di California. Sementara untuk di Eropa, negara yang sebagai area utama produsen Selada yaitu Italia, Spanyol, Prancis, Belanda, dan Inggris[3], [4].

### 2.3 Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra yang khususnya digunakan dalam komputer, menjadi citra yang memiliki kualitas yang lebih baik. Contohnya, dapat mengubah gambar yang sebelumnya gelap menjadi lebih terang, lalu akan dipertajam agar hasilnya lebih jelas. Pada umumnya, operasi pada pengolahan citra dapat diterapkan pada citra apabila:

1. Perbaikan citra perlu dilakukan untuk meningkatkan kualitas dari citra tersebut untuk menampilkan aspek informasi yang terdapat didalam citra.
2. Elemen yang terdapat dalam citra sangat perlu di kelompokkan.
3. Beberapa citra perlu penggabungan dengan bagian dari citra lainnya.[5]

Pada umumnya, pengolahan citra digital merujuk pada pemrosesan suatu gambar 2 (dua) dimensi yang menggunakan komputer. Dalam artian, pengolahan citra digital mengacu pada pemrosesan setiap data 2 (dua) dimensi. Citra digital merupakan suatu *array* yang berisi nilai – nilai, baik itu nilai *real* maupun kompleks yang direpresentasikan kedalam deretan bit tertentu.[6] Tahapan pada pengolahan citra digital meliputi:

1. Akuisisi Citra, untuk menentukan data yang diperlukan pada penelitian yang dikerjakan.
2. *Preprocessing*, untuk memastikan lancar atau tidaknya pada proses berikutnya.
3. Segmentasi, untuk membagi citra menjadi beberapa bagian yang mengandung informasi.
4. Representasi, untuk merepresentasikan wilayah sebagai daftar titik – titik koordinat dalam kurva dengan parameternya.
5. Deskripsi, dengan menggunakan ekstraksi ciri yang bertujuan untuk memilih informasi kuantitatif dengan ciri yang ada.
6. Pengenalan, untuk pemberian label pada objek yang disediakan.
7. Interpretasi, untuk pemberian makna pada objek – objek yang dikenali.
8. Basis Pengetahuan, memandu operasi dari masing – masing modul proses serta mengontrol interaksi pada modul – modul tersebut.[7]

### 2.4 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) menggunakan arsitektur khusus yang disesuaikan dengan sangat baik untuk mengklasifikasikan gambar. Dengan menggunakan arsitektur tersebut, membuat CNN sangat cepat untuk di latih. Pada saat ini, CNN lebih banyak digunakan dalam pengklasifikasian gambar.[8] Algoritma CNN adalah suatu *Multi-Layer Perceptron* (MLP) yang didesain khusus untuk mengidentifikasi citra atau gambar dua dimensi. Cara kerja *image recognition* pada CNN meniru cara kerja otak manusia untuk mengenali objek yang dijadikan *input*-an. Nantinya, komputer dapat membedakan berbagai objek dengan memiliki data latihnya sendiri. Perbedaan dari apa yang dilihat oleh manusia dengan apa yang dilihat oleh komputer adalah manusia melihat suatu objek itu seperti pada Gambar 2.1.



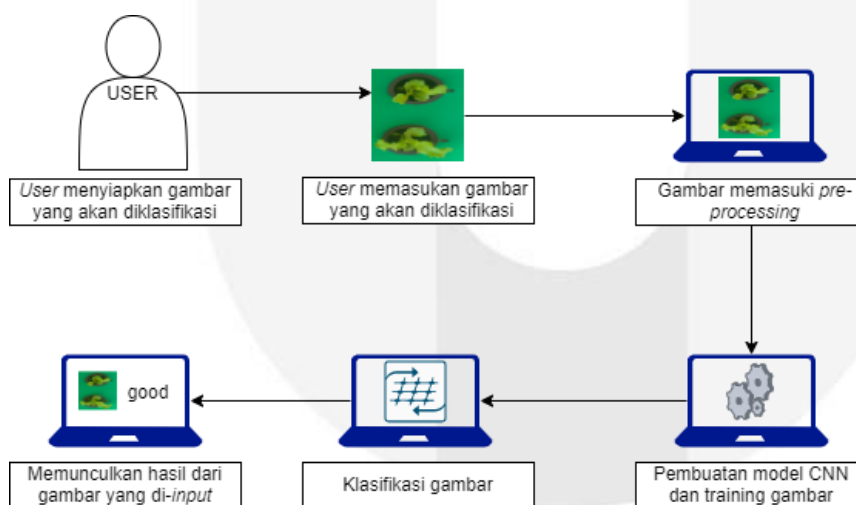
Gambar 2. 1 Gambar yang dilihat manusia

Sedangkan komputer melihat objek tersebut berupa *array* dari angka – angka yang merupakan nilai *pixel* dari objek tersebut. Bentuk *array* bergantung pada resolusi dan ukuran dari gambar tersebut. Contohnya adalah gambar yang memiliki format PNG dengan resolusi 360x360 piksel, maka dalam *array* menjadi 360x360x3 dengan angka 3 merupakan nilai dari Red Green Blue (RGB). Rentang nilai *array* dari 0 sampai 255 yang identik dengan intensitas *pixel* pada titik – titik tersebut. Nilai pada RGB dijadikan nilai *input* dan akan mengeluarkan *output* berupa nilai probabilitas. Secara sederhana, proses algoritma CNN dapat dijelaskan sebagai berikut:

1. Membaca *input* berupa gambar.
2. Memecah gambar menjadi beberapa gambar dengan ukuran yang lebih kecil.
3. Melakukan perhitungan algoritma CNN.
4. Melakukan klasifikasi.[9]

### 3. Metodologi Penelitian

#### 3.1. Gambaran Umum Sistem



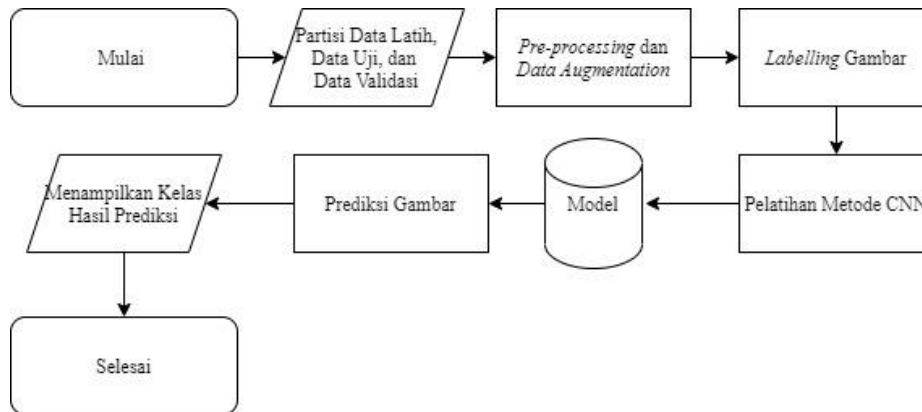
Gambar 3. 1 Model dari sistem

Berdasarkan Gambar 3.1, berikut adalah penjelasan dari tahapan – tahapan dari sistem:

1. Sistem akan dimulai setelah *user* telah menentukan gambar tanaman Selada mana yang akan diklasifikasi.
2. *User* memasukkan gambar tanaman Selada yang sudah di *crop* ke sistem untuk diklasifikasi.
3. Setelah di-*input*, gambar memasuki tahap *resize* dan *labelling* guna memudahkan proses *training* dan *klasifikasi*.
4. Setelah melewati *pre-processing*, sistem akan membuat model *layer* CNN sebanyak tiga *layer*.
5. Sistem melakukan *training* untuk melatih data latih yang sudah ada sehingga sistem bisa mengetahui mana yang kondisi baik dan mana yang tidak kondisi baik.

6. Sistem telah menyimpan hasil *training* tersebut dalam bentuk model yang otomatis terbuat dari hasil klasifikasi.
7. Sistem memunculkan hasil dari gambar yang telah di-*input* oleh *user* dengan memunculkan gambar tanaman Selada dan label dari tanaman Selada tersebut (bad atau good).

### 3.2 Perancangan Sistem



Gambar 3. 2 Flowchart Diagram

Flowchart diagram yang digunakan pada Gambar 3.2 menjelaskan bagaimana perancangan sistem ini berjalan. Berikut adalah penjelasan dari perancangan sistem tersebut:

1. Sistem dimulai dengan *user* telah menyiapkan tanaman Selada yang akan digunakan sebagai penelitian.
2. *User* memilah gambar untuk digunakan sebagai data latih, data uji, dan data validasi.
3. Setelah gambar terdeteksi pada sistem, sistem akan melakukan *pre-processing* yaitu *resize* gambar menjadi ukuran 250x250 piksel dan sistem juga melakukan *Data Augmentation* untuk mengenali gambar lebih dalam.
4. Sistem memberikan label pada gambar yang telah diproses tadi dengan memilah menjadi dua kelas, kondisi tidak baik dan kondisi baik.
5. Gambar telah masuk pemodelan dari metode CNN, dimana gambar akan disesuaikan prosesnya menurut *layer* yang telah ditentukan.
6. Model *layer* telah dibuat untuk digunakan sebagai klasifikasi gambar pada data gambar.
7. Sistem melakukan *training* untuk mengenali gambar serta mengklasifikasikan gambar tersebut termasuk ke dalam kondisi tidak baik atau baik.
8. Sistem membaca gambar yang telah di-*input*.
9. Sistem menampilkan hasil dari gambar yang telah di-*input* apakah kondisi tanaman Selada tersebut termasuk ke dalam kelas kondisi tidak baik atau kelas kondisi baik.

### 3.3 Spesifikasi Proses

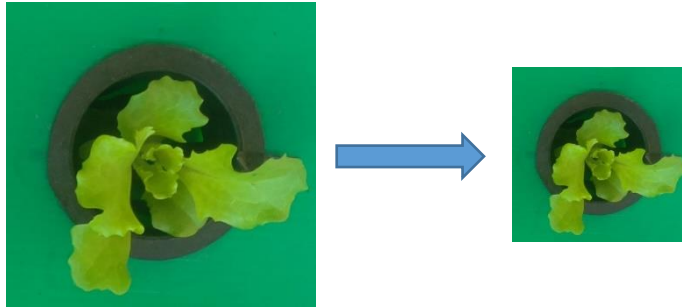
#### 3.3.1 Proses Pengambilan Gambar

Pada tahap pertama ini, *user* harus menyiapkan berbagai data untuk diolah kembali sebagai keperluan data latih dan data uji, serta sampel gambar untuk dijadikan acuan terhadap hasil akhir nantinya. Dalam pengambilan gambar ini, penempatan kamera tepat berada diatas tanaman Selada yang ditanam pada media tanam Kit Hidroponik. Sudut yang ditempatkan kamera terhadap media tanam tersebut yaitu 90 derajat dengan jarak 30cm. Proses pengambilan gambar tersebut dibantu oleh mikro komputer yaitu Raspi dan kamera yang saling terhubung. Hasil gambar dari tanaman Selada tersebut akan diolah lagi oleh *user* pada proses selanjutnya.

#### 3.3.2 Proses Pengolahan Gambar

Pada tahap ini, *user* harus melakukan *cropping* secara manual, untuk data tersebut digunakan pada data latih dan data uji. Proses *cropping* secara manual ini bertujuan untuk memisahkan tanaman Selada yang memiliki kondisi tidak baik dengan kondisi baik serta menganalisa dari perbedaan dua kondisi tersebut. Perbedaan dari dua kondisi tersebut akan digunakan sebagai penentuan kelas gambar yang nantinya akan di-*input*. Proses *cropping* ini hanya mengambil satu tanaman Selada dalam satu *frame*, untuk menghindari asumsi ganda. Apabila data latih dan data uji memiliki lebih dari satu tanaman dalam satu *frame* serta masukan dari gambar tersebut memiliki kriteria dua kelas dalam satu *frame*, maka hasil akhir dari sistem tersebut akan menimbulkan ambigu, karena sistem akan membaca kondisi tanaman yang tidak baik.

Proses pengolahan gambar juga terjadi dalam sistem yang dibuat oleh penulis. Proses tersebut adalah *resize* yang berfungsi untuk mengecilkan ukuran gambar menjadi ukuran 250 x 250 piksel. Ukuran ini menurut penulis cukup untuk diproses dalam sistem, mengingat ukuran asli dari gambar yang akan diproses dalam data latih dan data uji cukup besar.



Gambar 3. 3 Proses *resize* pada sistem

Setelah gambar memasuki proses *resize*, gambar akan langsung memasuki proses *Data Augmentation* gambar pada *generator*. Proses ini akan mengandalkan sistem tersebut untuk membaca gambar lebih dalam lagi, supaya sistem dapat mengenal lebih baik pada gambar tersebut. *Data Augmentation* ini pun bertujuan untuk membedakan tanaman Selada yang ingin diklasifikasi dengan *background* yang ada pada gambar tersebut, baik di dalam data latih maupun pada data uji. Hal yang dikerjakan oleh generator ini seperti melebarkan gambar tersebut ke berbagai sisi dan memanipulasi gambar dengan tanpa menghilangkan inti dari gambar yang akan diklasifikasi.



Gambar 3. 4 Beberapa contoh proses *Data Augmentation*

Gambar yang sudah di melewati proses *Data Augmentation*, akan memasuki proses *labeling*. Proses ini bertujuan untuk memberikan label pada data latih dan data uji yang akan diproses. Pada sistem ini, gambar akan dipisah menjadi dua kelas yaitu tanaman Selada dengan kondisi tidak baik dengan kondisi baik. Kelas tanaman Selada dengan kondisi tidak baik akan dikategorikan sebagai "0 : bad". Sedangkan kelas tanaman Selada dengan kondisi baik akan dikategorikan sebagai "1 : good".



Gambar 3. 5 Pemberian label pada gambar sesuai dengan kelas yang ditentukan

Setelah gambar diberi label, sistem akan membuat model *layer*. *Layer* yang digunakan penulis ini terdiri dari:

1. *Convolutional 2D*  
*Layer Convolutional 2D* atau Conv2D dengan ReLU sebagai *activation layer* ini memiliki inputan (250, 250, 3) yang berarti ukuran gambar 250 x 250 piksel dengan warna RGB. Conv2D berfungsi untuk mempertajam serta meningkatkan performa dari CNN.
2. *Flatten*  
*Layer Flatten* yang berfungsi untuk membuat semua matriks yang ada pada gambar menjadi berukuran *single vector* yang nanti hasilnya akan menjadi *input* bagi *layer* selanjutnya,
3. *Dense*  
*Layer Dense* yang sebagai *output layer*. Pada kasus ini, *layer* menggunakan *activation layer softmax*.

Setelah pembuatan model *layer* CNN, sistem akan meng-*compile* keseluruhan *layer* tersebut dengan optimizer dari 'adam', loss menggunakan 'categorical\_crossentropy', serta metrics oleh 'accuracy'. Selanjutnya, sistem akan masuk ke tahap training pada gambar yang sudah terbaca oleh sistem.

### 3.3.3 Proses Klasifikasi Gambar

Setelah proses training serta test dari sistem, sistem akan memperhitungkan akurasi dalam bentuk Confusion Matrix. Confusion Matrix merupakan sebuah tabel yang digunakan untuk mendeskripsikan suatu performa dari model klasifikasi pada sekumpulan data uji yang nilai sebenarnya diketahui. Ini memungkinkan visualisasi kinerja suatu algoritma. Pada akhir pengujian didapat suatu nilai sebagai berikut:

```

[[15  0]
 [ 0 15]]
Classification Report
      precision    recall  f1-score   support

   bad         1.00      1.00      1.00        15
   good         1.00      1.00      1.00        15

 avg / total         1.00      1.00      1.00        30

```

Gambar 3. 6 Gambar Tabel *Confusion Matrix* pada pengujian akhir sistem

#### 1. Accuracy

Merupakan rasio prediksi True (berlaku untuk positif dan negatif) dengan keseluruhan data. Didapat suatu perhitungan seperti di bawah ini:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{15 + 15}{15 + 0 + 0 + 15} = 1$$

#### 2. Recall

Merupakan rasio prediksi True Positive yang dibandingkan dengan keseluruhan data yang True Positive. Didapat suatu perhitungan seperti di bawah ini:

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{15}{15 + 0} = 1$$

#### 3. Precision

Merupakan rasio prediksi True Positive yang dibandingkan dengan keseluruhan hasil yang telah diprediksi Positive. Didapat suatu perhitungan seperti di bawah ini:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{15}{15 + 0} = 1$$

#### 4. F1 - Score

Merupakan perbandingan rata-rata dari Precision dan Recall yang dibobotkan. Didapat suatu perhitungan seperti di bawah ini:

$$\text{F1 - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times 1 \times 1}{1 + 1} = 1$$

Setelah melewati proses perhitungan Confusion Matrix, sistem akan membuat dan menyimpan hasil keseluruhan sistem di dalam sebuah model yang nantinya akan dipanggil oleh sistem untuk menentukan tanaman Selada yang telah di masukkan ke dalam sistem, apakah tanaman Selada tersebut termasuk kedalam kelas kondisi tidak baik atau kondisi baik.

## 4. Implementasi dan Pengujian

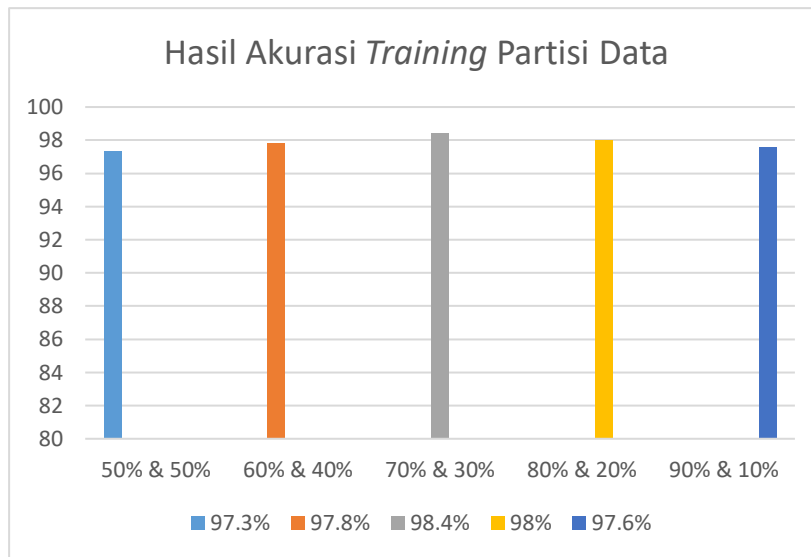
### 4.1 Pengujian Partisi Data

Pengujian partisi data berfungsi untuk mengetahui banyak data yang digunakan pada tahap pengujian selanjutnya menggunakan banyak data yang mana. Pada penelitian ini memanfaatkan gambar tanaman Selada sebanyak 300 dengan pembagian kelas kondisi tidak baik dengan kelas kondisi baik yang sama rata dengan jumlah 150 gambar tanaman Selada.

Pengujian partisi data memanfaatkan nilai *default* atau bawaan dari sistem yaitu *learning rate* pada nilai 0.001, jumlah *epochs* 100, dan *batch size* 10. Pengujian dilakukan dengan pembagian persentase partisi data sebagai berikut:

1. Pengujian dengan data latih sebanyak 270 gambar tanaman Selada (90%) dan data uji sebanyak 30 gambar tanaman Selada (10%).
2. Pengujian dengan data latih sebanyak 240 gambar tanaman Selada (80%) dan data uji sebanyak 60 gambar tanaman Selada (20%).
3. Pengujian dengan data latih sebanyak 210 gambar tanaman Selada (70%) dan data uji sebanyak 90 gambar tanaman Selada (30%).
4. Pengujian dengan data latih sebanyak 180 gambar tanaman Selada (60%) dan data uji sebanyak 120 gambar tanaman Selada (40%).

5. Pengujian dengan data latih sebanyak 150 gambar tanaman Selada (50%) dan data uji sebanyak 150 gambar tanaman Selada (50%).



Gambar 4. 1 Grafik pengujian akurasi *training* pada partisi data

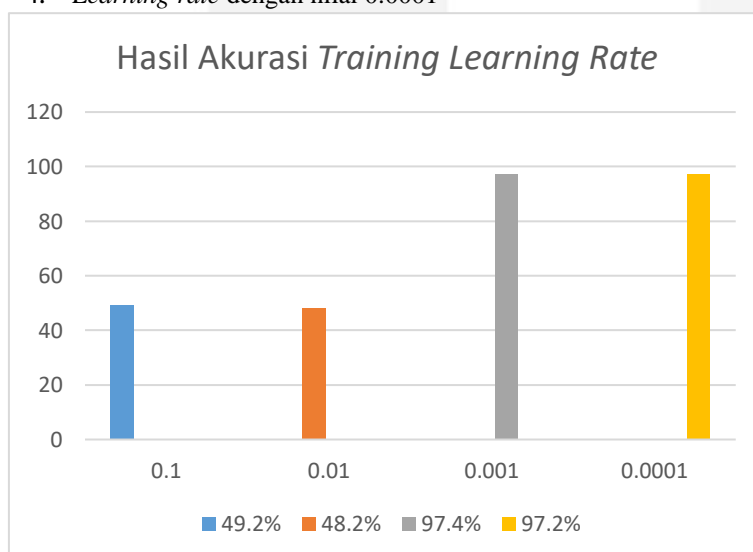
Dari hasil pengujian partisi data, didapatkan tingkat akurasi tertinggi pada perhitungan *confusion matrix*-nya dengan pembagian partisi data 90% untuk data latih dan 10% untuk data uji dengan nilai akurasi yaitu 97.6%. Dengan hasil tersebut, maka telah ditetapkan untuk pengujian selanjutnya dengan menggunakan data latih sebanyak 90% dan data uji sebanyak 10% dari data gambar tanaman Selada yang ada.

#### 4.2 Pengujian *Learning Rate*

Pengujian dengan parameter *learning rate* berfungsi untuk mengetahui pengaruh jumlah *learning rate* terhadap akurasi yang akan muncul pada sistem. *Learning rate* merupakan suatu tingkat ketelitian pada sistem untuk memperbaiki nilai *error* atau pun untuk meminimalisir nilai dari nilai *loss*.

Pengujian parameter *learning rate* menggunakan data latih berupa data gambar tanaman Selada yang sudah ada. *Learning rate* yang diuji memanfaatkan dari nilai akurasi maksimal dari pengujian partisi data dengan data latih 90% dan data uji 10%, jumlah *epochs* 100, dan *batch size* 10. Pengujian dilakukan *learning rate* dengan nilai sebagai berikut:

1. *Learning rate* dengan nilai 0.1
2. *Learning rate* dengan nilai 0.01
3. *Learning rate* dengan nilai 0.001
4. *Learning rate* dengan nilai 0.0001



Gambar 4. 2 Grafik pengujian akurasi *training* pada *learning rate*

Dari hasil pengujian *learning rate* didapatkan nilai akurasi tertinggi pada nilai 0.001. Tetapi, dengan pertimbangan atas hasil dari *confusion matrix*, nilai 0.0001 lebih di unggulkan karena memiliki akurasi paling

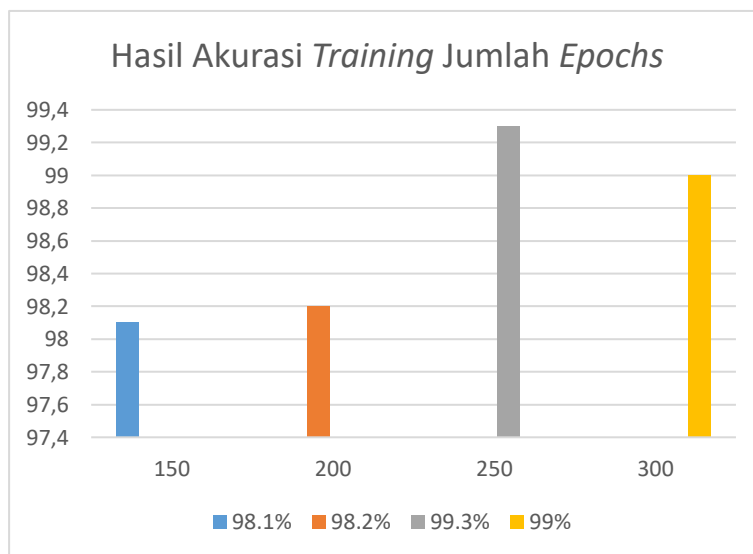
tinggi dibanding nilai *learning rate* lainnya. Oleh karena itu, untuk pengujian selanjutnya yaitu pada pengujian jumlah *epochs* akan menggunakan partisi data 90% untuk data latih dan 10% untuk data uji dan nilai *learning rate* 0.0001.

#### 4.3 Pengujian Epochs

Pengujian dengan parameter *epochs* berfungsi untuk mengetahui pengaruh jumlah *epochs* terhadap akurasi yang akan muncul pada sistem. *Epochs* merupakan suatu banyak tahapan yang diperlukan bagi sistem dalam melakukan proses *training*.

Pengujian parameter *epochs* menggunakan data latih berupa data gambar tanaman Selada yang sudah ada. *Epochs* yang diuji menggunakan nilai akurasi maksimal dari pengujian partisi data dengan data latih 90% dan data uji 10%, *learning rate* 0.0001, dan *batch size* 10. Pengujian dilakukan *epochs* dengan jumlah sebagai berikut:

1. *Epochs* dengan jumlah 150
2. *Epochs* dengan jumlah 200
3. *Epochs* dengan jumlah 250
4. *Epochs* dengan jumlah 300



Gambar 4. 3 Grafik pengujian akurasi *training* pada jumlah *epochs*

Dari hasil pengujian jumlah *epochs*, didapatkan tingkat akurasi tertinggi dengan pembagian jumlah *epochs* 250 dengan nilai akurasi yaitu 99.3%. Akurasi tersebut pun didukung dengan nilai *confusion matrix* yang tinggi. Dengan hasil tersebut, maka telah ditetapkan untuk pengujian selanjutnya dengan menggunakan partisi data 90% untuk data latih dan 10% untuk data uji, nilai *learning rate* 0.0001, dan jumlah *epochs* 250.

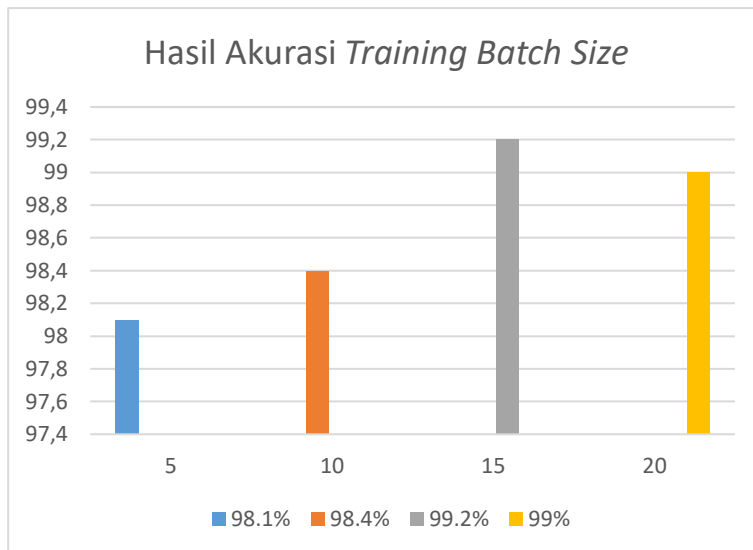
#### 4.4 Pengujian Batch Size

Pengujian dengan *batch size* berfungsi untuk menentukan jumlah sampel yang nantinya dikerjakan sebelum memperbarui parameter pada model *Neural Network*. *Batch* itu sendiri merupakan pembagian sebuah *dataset* menjadi atau satu set, bahkan suatu bagian.

Pengujian *batch size* ini menggunakan data latih berupa gambar tanaman Selada yang sudah ada. *Batch size* yang diuji menggunakan nilai akurasi maksimal dari pengujian partisi data dengan data latih 90% dan data uji 10%, *learning rate* 0.0001, dan jumlah *epochs* 250. Pengujian dilakukan dengan nilai *batch size* sebagai berikut:

1. *Batch size* dengan nilai 5
2. *Batch size* dengan nilai 10
3. *Batch size* dengan nilai 15
4. *Batch size* dengan nilai 20





Gambar 4. 4 Grafik pengujian akurasi *training* pada *batch size*

Dari hasil pengujian nilai *batch size*, didapatkan nilai akurasi tertinggi dengan nilai *batch size* 15 dengan hasil akurasinya sebesar 99.2%. Akurasi tersebut pun memiliki nilai *confusion matrix* yang tinggi.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

1. Perancangan sistem yang dapat mendeteksi kondisi dari daun tanaman Hidroponik khususnya tanaman Selada dengan menggunakan metode *Convolutional Neural Network* (CNN), dapat berfungsi dengan baik.
2. Dengan menggunakan 300 gambar tanaman Selada, yang di partisi menjadi 90% untuk data latih dan 10% untuk data uji, sistem dapat berjalan dengan baik.
3. Sistem klasifikasi kondisi tanaman Selada ini memiliki dua kelas yang terdiri dari kondisi tidak baik dan kondisi baik. Dengan menggunakan algoritma *Convolutional Neural Network* (CNN), hasil performansi atau akurasi terbaik yang dikeluarkan dengan menggunakan partisi data 90% untuk data latih dan 10% untuk data uji, *learning rate* sebesar 0.0001, jumlah *epochs* 250, dan nilai *batch size* 15 mencapai 99.2%.

### 5.2 Saran

1. Penelitian selanjutnya diharapkan dapat mendeteksi lebih dari satu tanaman dalam satu *frame*.
2. Penelitian selanjutnya diharapkan dapat menambahkan kelas lebih banyak lagi, supaya pendeteksian terhadap kondisi tanaman Selada dapat lebih detail.
3. Diharapkan tampilan antarmuka pada penelitian selanjutnya, dapat lebih *user friendly* untuk memudahkan *user* yang menggunakan aplikasi tersebut.
4. Pada penelitian selanjutnya, diharapkan dapat memvariasikan jarak deteksi dari pendeteksian kondisi tanaman Selada.
5. Sistem selanjutnya diharapkan dapat diimplementasikan ke Green House dari skala rumahan (kecil) sampai skala industri (besar).

### Daftar Pustaka:

- [1] S. Liu, J. Zhang, H. Zhang, N. Chen, Z. Chen, and T. Li, "A device designed for plant growth with automatic adjustment," *2018 Int. Conf. Electron. Technol. ICET 2018*, pp. 442–445, 2018.
- [2] A. Hamda, "Hidroponik, alternatif atas defisit sumberdaya," *Medium*. [Online]. Available: [https://medium.com/@amik\\_62838/hidroponik-sebuah-jawaban-80d3f8bbf338](https://medium.com/@amik_62838/hidroponik-sebuah-jawaban-80d3f8bbf338).
- [3] T. N. (Managing E. H. L. J. M. W. Widholm, J M and J. M. Widholm, "Biotechnology in Agriculture and Forestry," *Structure*, vol. 62, pp. 2–17, 2005.
- [4] M. R. Davey, P. Anthony, P. Van Hooff, J. B. Power, and K. C. Lowe, "III . 4 Lettuce," *Biotechnol. Agric. For.*, vol. 59, pp. 221–249, 2007.
- [5] -, "Pengantar Pengolahan Digital," in *Pengolahan Digital*, .
- [6] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: C.V ANDI OFFSET (Penerbit Andi), 2010.
- [7] E. Mulyanto, *Teori Pengolahan Citra Digital*, Edisi 1. Yogyakarta: C.V ANDI OFFSET (Penerbit Andi), 2009.
- [8] M. Nielsen, "Neural Networks and Deep Learning."
- [9] R. Primartha, *Belajar Machine Learning*. Bandung: Informatika Bandung, 2018.