

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Prinsip Kerja Program

Prinsip kerja program yaitu dengan melakukan pra pengolahan citra terhadap foto fisik dari permukaan buah manggis agar ukuran seluruh data citra seragam untuk digunakan sebagai data penelitian. Kumpulan citra tersebut dibagi menjadi citra latih dan citra uji dengan data set yang ditentukan berdasarkan *4-fold Cross Validation*, metode *Convolutional neural network* dari *deep learning* digunakan untuk melakukan proses latih dan uji sehingga *hyperplane* membentuk area klasifikasi. Dalam tahap uji, CNN diimplementasikan dengan lima lapisan agar mencapai akurasi yang lebih tinggi. Pada tahap uji, data yang baru akan dipetakan pada model area latih dan hasilnya bergantung pada data tersebut akan terletak pada area tertentu sehingga terklasifikasi.

4.2 Hasil Pengambilan Data

Pengambilan data citra dilakukan menggunakan kamera digital dengan pengaturan resolusi kamera dan jarak serta pencahayaan yang sama. Pengambilan data citra ini dilakukan di Laboratorium Pasca Panen, Fakultas Pertanian, Universitas Muhammadiyah Yogyakarta yang selanjutnya diserahkan pada Fakultas Teknik untuk dilakukan penelitian.

Data citra yang telah diperoleh lalu di klasifikasi kualitas permukaannya secara manual. Untuk mengklasifikasikan sampel data citra yang cacat maupun citra tidak cacat ini menggunakan pengamatan secara langsung atau secara visual yang nantinya akan dijadikan perbandingan dengan hasil klasifikasi program. Dari

klasifikasi kualitas permukaan buah manggis yang telah dilakukan dipilih sebanyak 120 citra untuk dijadikan bahan penelitian. Dari data total 120 citra tersebut merupakan gabungan dari citra latih dan citra uji, dimana ke-120 citra yang dihasilkan tersebut terdiri 30 citra cacat dan 90 citra tidak cacat dari citra permukaan buah manggis, kemudian sampel citra tersebut nantinya akan dibagi kedalam 4 kelas uji yaitu kelas *fold-1*, *fold-2*, *fold-3* dan *fold-4*, dalam satu *fold* terdapat data uji yang terdiri atas 7-8 citra cacat dan 22-23 citra tidak cacat. Sedangkan untuk data latih merupakan sisa dari citra cacat dari masing-masing *fold* yang tidak termasuk dalam data uji dari ke 120 data citra tersebut, yang terdiri atas 22-23 citra cacat dan 67-68 citra tidak cacat. Pembagian kelas pada citra bertujuan untuk mempermudah proses validasi pada proses klasifikasi data sehingga menghasilkan akurasi yang maksimal. Data citra beserta pembagian kelasnya dapat dilihat pada Lampiran 2 Citra Uji dan Citra Latih.

Tabel 4.1 Data citra buah manggis

Kelas	Jumlah Citra	Keterangan	
		Uji	Latih
<i>fold-1</i>	120	8 cacat	22 cacat
		22 tidak cacat	68 tidak cacat
<i>fold-2</i>	120	7 cacat	23 cacat
		23 tidak cacat	67 tidak cacat
<i>fold-3</i>	120	7 cacat	23 cacat
		23 tidak cacat	67 tidak cacat
<i>fold-4</i>	120	8 cacat	22 cacat
		22 tidak cacat	68 tidak cacat

4.3 Hasil Perancangan Program

4.3.1 Akuisisi Data

Data citra diambil menggunakan kamera digital dengan jarak, pencahayaan, serta resolusi gambar yang sama yaitu sebesar 6000 x 4000 piksel. Pada pengambilan data, dalam satu foto berisi dua buah manggis seperti yang ditunjukkan pada Gambar 4.1. Kemudian dilakukan *cropping* foto agar menjadi foto satu buah manggis.



Gambar 4.1 Sampel awal pengambilan data citra

4.3.2 Labeling Citra

Pemberian label terhadap citra dilakukan setelah proses akuisisi data, nama setiap citra diubah berdasarkan kondisi permukaannya, citra cacat diberi nama defect sedangkan citra tidak cacat diberi nama fine. Format penamaan citra ini berdasarkan kondisi permukaan citra tersebut kemudian diikuti urutan citra. Contoh penamaan untuk citra cacat adalah defect1.jpg sedangkan untuk citra tidak retak adalah fine1.jpg, hingga nomor urut setelah nama kondisi mencapai 30 defect dan 90 fine. Setelah tahap *labeling* dari 120 citra uji dibagi ke dalam 4-*fold* dengan masing – masing 30 citra latih pada setiap *fold*-nya yang terdiri dari 7-8

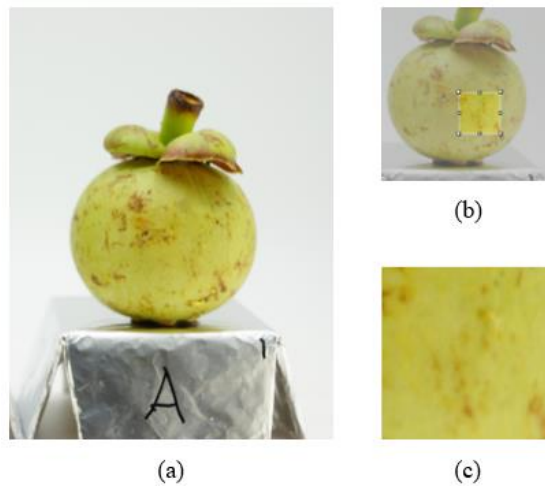
citra cacat dan 22-23 citra tidak cacat. Sedangkan untuk citra latih merupakan citra-citra yang tidak termasuk dalam citra uji di setiap masing-masing *fold*, yang setiap *fold* nya terdiri atas 22-23 citra cacat dan 67-68 citra tidak cacat.

Tabel 4.2 Detail pembagian citra uji

Kelas	Pembagian Citra Cacat	Pembagian Citra Tidak Cacat	Total citra
<i>fold-1</i>	defect1.jpg s/d defect8.jpg	fine1.jpg s/d fine22.jpg	30
<i>fold-2</i>	defect9.jpg s/d defect15.jpg	fine23.jpg s/d fine45.jpg	30
<i>fold-3</i>	defect16.jpg s/d defect22.jpg	fine46.jpg s/d fine68.jpg	30
<i>fold-4</i>	defect23.jpg s/d defect30.jpg	fine69.jpg s/d fine90.jpg	30

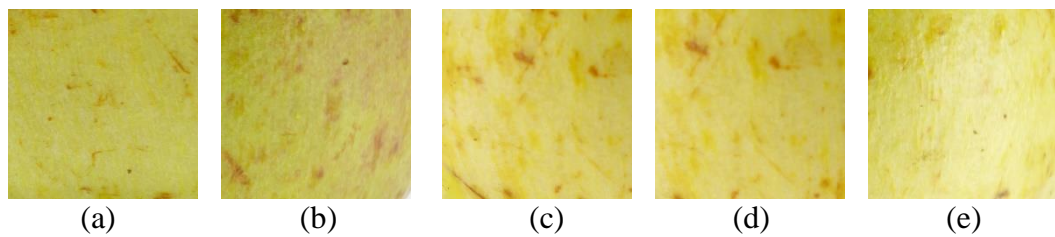
4.3.3 Pra Pengolahan Citra

Citra hasil pengambilan data memiliki resolusi yang sama yaitu 6000 x 4000 piksel, namun dalam penelitian ini data citra yang digunakan adalah bagian permukaan yang terlihat jelas pada foto permukaan buah manggis. Untuk itu dilakukan *cropping* menggunakan aplikasi PhotoScape v3.6.5. Ukuran standar data citra pada penelitian ini adalah 512 x 512 piksel. Sehingga seluruh data foto yang akan digunakan untuk penelitian ini melalui tahap proses *cropping* sehingga seluruh data citra ukurannya seragam. setelah dilakukan proses *cropping* selanjutnya akan disamakan ukuran sebesar 512 x 512 piksel. Proses ini dilakukan untuk mempermudah pengambilan data citra agar lebih akurat. Hal ini sangat berpengaruh selain untuk mereduksi waktu komputasi juga menjadikan ukuran citra seragam. Proses *cropping* ditunjukkan pada gambar 4.2.

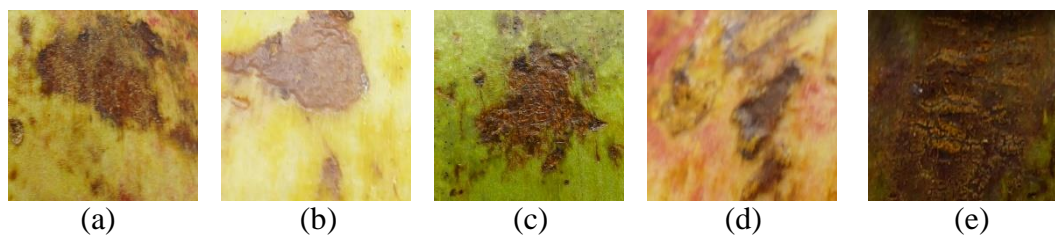


Gambar 4.2 (a) citra awal sebelum pengolahan (b) diresize menjadi 512x512 piksel (c) sehingga menghasilkan citra permukaan manggis

Setelah dilakukan proses *cropping*, proses selanjutnya adalah memilah data citra yang cocok untuk dijadikan bahan penelitian serta memisahkan permukaan citra yang dianggap cacat dan tidak cacat dengan pengamatan secara langsung atau secara visual. Beberapa sampel hasil proses *cropping* dari citra yang akan menjadi bahan penelitian ditunjukkan pada Gambar 4.3 dan Gambar 4.4



Gambar 4.3 Data citra tidak cacat hasil proses *cropping* 512x512



Gambar 4.4 Data citra cacat hasil proses *cropping* 512x512

4.3.4 Pengujian Parameter

Pada proses pengujian parameter ini, data set yang digunakan untuk pengujian adalah *fold 4*. *Fold 4* dijadikan sebagai data set pengujian parameter dikarenakan hasil akurasi proses latih dan uji pada *fold 4* paling tidak stabil dibandingkan dengan *fold* lainnya. Sehingga *fold 4* dijadikan acuan optimalisasi program dengan mengubah parameter-parameternya.

a. Parameter variabel Layer

Pengujian parameter ini dilakukan untuk menghasilkan variabel *Layer* mana yang terbaik. Pada pengujian ini menggunakan pengaturan fungsi `trainingOptions` secara *default* yaitu menggunakan `MaxEpoch 15` dengan `2` `MiniBatchSize`. Parameter-parameter yang diujikan yaitu `convolution2dLayer`, `maxPooling2dLayer`, dan `reluLayer`. Pada `convolution2dLayer` parameter yang digunakan adalah `convolution2dLayer(2,10)` dan `convolution2dLayer(5,20)`. Sedangkan pada `maxPooling2dLayer` parameter yang digunakan adalah `maxPooling2dLayer(2,'Stride',2)` dan `maxPooling2dLayer(4,'Padding',2)`. Pada `reluLayer` parameter yang diujikan adalah menggunakan atau tidak menggunakan parameter `reluLayer`. Hasil pengujian parameter fungsi `trainingOptions` ditunjukkan pada tabel 4.3 untuk hasil tanpa `reluLayer` dan tabel 4.4 untuk hasil yang menggunakan parameter `reluLayer`.

Tabel 4.3 Hasil Pengujian Tanpa reluLayer

Conv \ Pool	(2, S2)	(4, P2)
(2,10)	90%	80%
(5,20)	86,67%	70%

Tabel 4.4 Hasil Pengujian Dengan reluLayer

Conv \ Pool	(2, S2)	(4, P2)
(2,10)	70%	76,67%
(5,20)	70%	76,67%

Pada tabel 4.3 dan tabel 4.4 menunjukkan hasil pengujian dari variabel Layer. Hasil pengamatan pada tabel 4.3 dan tabel 4.4 menunjukkan parameter `convolution2dLayer` dengan ukuran panjang lebar *filter* 2 dan angka *filter* 10 serta `maxPooling2dLayer` dengan ukuran *pool* 2 dan *Stride* 2 memiliki akurasi yang tertinggi dari parameter yang lain dengan akurasi yang optimal mencapai 90%. Sehingga parameter pada variabel Layer menggunakan `convolution2dLayer(2,10)` yang artinya layer konvolusi nya menggunakan ukuran panjang dan lebar filter bernilai 2 serta 10 adalah jumlah dari filternya, sedangkan `maxPooling2dLayer(2,'Stride',2)` yang artinya memiliki ukuran panjang dan lebar dari daerah poolnya bernilai 2 serta 2 *Stride* berarti ukuran langkah untuk melintasi input secara vertikal dan horizontal sebanyak 2 langkah, namun dengan akurasi ini tidak menggunakan `reluLayer`. Gambar 4.5 menunjukkan hasil dari proses pengujian dengan akurasi yang tertinggi. Untuk

hasil dari proses pengujian lain dapat dilihat di Lampiran 1 Hasil Pengujian Parameter.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
2	50	3.74	-0.0000	100.00%	0.000100
3	100	7.40	-0.0000	100.00%	0.000100
4	150	11.06	7.9712	50.00%	0.000100
5	200	14.72	-0.0000	100.00%	0.000100
6	250	18.38	-0.0000	100.00%	0.000100
7	300	22.05	-0.0000	100.00%	0.000100
8	350	25.71	-0.0000	100.00%	0.000100
9	400	29.37	7.9712	50.00%	0.000100
10	450	33.07	7.9712	50.00%	0.000100
12	500	36.79	-0.0000	100.00%	0.000100
13	550	40.52	-0.0000	100.00%	0.000100
14	600	44.24	7.9712	50.00%	0.000100
15	650	47.96	-0.0000	100.00%	0.000100

accuracy =

0.9000

Gambar 4.5 Hasil proses pengujian parameter layer dengan akurasi tertinggi

b. Parameter fungsi trainingOptions

Setelah mendapatkan parameter yang terbaik dari proses pengujian parameter variabel Layer, maka dilakukan pengujian parameter fungsi trainingOptions mana yang terbaik. Seperti pada proses pengujian parameter variabel Layer, data set yang digunakan yaitu *fold 4*. Dari hasil yang didapatkan setelah melakukan proses pengujian parameter variabel Layer, parameter dengan hasil yang paling optimal didapatkan dengan menggunakan layer convolution2dLayer dengan ukuran panjang lebar *filter 2* dan angka *filter 10* serta maxPooling2dLayer dengan ukuran *pool 2* dan *Stride 2*. Jadi secara *default* pengaturan tersebut digunakan untuk menguji parameter-parameter pada fungsi trainingOptions. Parameter-parameter yang akan diujikan yaitu MaxEpoch dengan nilai 15, 30, dan 50 serta MiniBatchSize dengan nilai 2, 5,

dan 10. Hasil pengujian parameter fungsi `trainingOptions` ditunjukkan pada tabel 4.5.

Tabel 4.5 Hasil Pengujian Parameter Fungsi

Batch \ Epoch	2	5	10
15	90%	86,67%	76,67%
30	70%	70%	60%
50	86,67%	96,67%	70%

Pada tabel 4.5 menunjukkan hasil pengujian dari fungsi `trainingOptions`. Hasil pengamatan pada tabel 4.5 menunjukkan parameter `MaxEpoch` dengan nilai 50 serta parameter `MiniBatchSize` dengan nilai 5 memiliki akurasi yang tertinggi dari parameter yang lain dengan akurasi yang optimal mencapai 93,33%. Sehingga parameter pada fungsi `trainingOptions` menggunakan `MaxEpoch` 50 dengan 5 `MiniBatchSize`, yang artinya setiap proses latih melalui lima puluh kali proses *feedforward* dan *backpropagation* dengan lima *batch* dan jumlah keseluruhan iterasi adalah 900. Gambar 4.6 menunjukkan hasil dari proses pengujian dengan akurasi yang tertinggi. Untuk hasil dari proses pengujian lain dapat dilihat di Lampiran 1 Hasil Pengujian Parameter.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
3	50	7.51	1.5230	80.00%	0.000100
6	100	15.05	-0.0000	100.00%	0.000100
9	150	22.60	3.1885	80.00%	0.000100
12	200	30.15	-0.0000	100.00%	0.000100
14	250	37.72	-0.0000	100.00%	0.000100
17	300	45.32	3.1885	80.00%	0.000100
20	350	52.86	-0.0000	100.00%	0.000100
23	400	60.39	3.1885	80.00%	0.000100
25	450	67.95	3.1885	80.00%	0.000100
28	500	75.50	-0.0000	100.00%	0.000100
31	550	83.06	-0.0000	100.00%	0.000100
34	600	90.61	3.1885	80.00%	0.000100
37	650	98.16	-0.0000	100.00%	0.000100
39	700	105.72	-0.0000	100.00%	0.000100
42	750	113.27	3.1885	80.00%	0.000100
45	800	120.82	-0.0000	100.00%	0.000100
48	850	128.36	3.1885	80.00%	0.000100
50	900	135.91	3.1885	80.00%	0.000100

accuracy =

0.9667

Gambar 4.6 Hasil proses pengujian parameter dengan akurasi tertinggi

4.3.5 Proses Latih

Proses latih dalam penelitian ini dilakukan untuk memperoleh akurasi yang tinggi ketika program dilakukan proses uji. Dalam tahap latih, jumlah *layer* yang digunakan terdiri atas *image input layer* sebagai input, *convolution layer*, *subsampling layer*, *fully connected layer*, *softmax layer* sebagai layer aktivasi, serta *classification output layer* sebagai output. Selain itu dalam penelitian ini menggunakan parameter pengaturan *training* yang sama yaitu nilai maksimum *epoch* adalah 50 dan ukuran *mini batch* adalah 5. Sedangkan untuk pengaturan parameter layer *convolution2dLayer* dengan ukuran panjang lebar *filter* 2 dan angka *filter* 10 serta *maxPooling2dLayer* dengan ukuran *pool* 2 dan *Stride* 2

Dalam penelitian ini, proses latih dilakukan menggunakan 4 *fold*. Setiap *fold* memiliki data set latih, di mana data citra tersebut adalah citra dari *fold* selain dari *fold* itu sendiri (*fold* data uji). Misalnya adalah ketika *fold* 1 dijadikan data uji,

maka *fold 2*, *fold 3*, dan *fold 4* menjadi data latih untuk *fold 1*. Setiap *fold* memiliki 120 citra di mana dari 120 citra itu dibagi menjadi 30 data uji dan 90 data latih. Setiap *fold* melalui proses tahapan 1 kali latihan sekaligus pengujian. Hal ini dilakukan untuk membandingkan setiap proses pelatihan dan menganalisa perkembangan akurasi yang didapatkan dari proses uji tiap masing-masing *fold*. Pengklasifikasian yang dilakukan pada proses uji merupakan hasil dari proses pembelajaran pada tahap latih.

a. Hasil latih *fold 1*

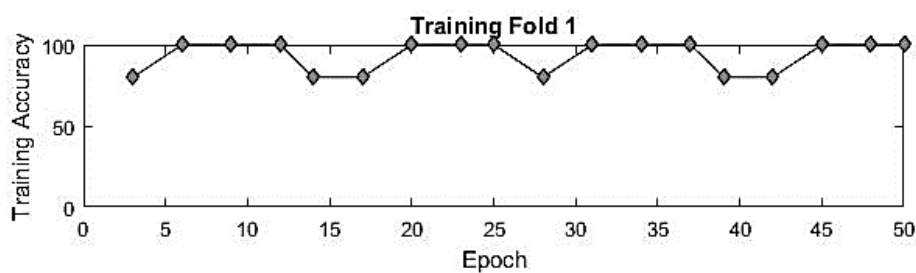
Di dalam data set *fold 1*, terdapat 30 citra latih di mana terdapat 8 citra cacat dan 22 citra tidak cacat. Sementara citra sisanya yaitu 90 akan dijadikan sebagai citra latih dengan 22 citra cacat dan 68 citra tidak cacat. Hasil akurasi training dari *fold 1* dapat dilihat pada gambar 4.7.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
3	50	7.54	3.1885	80.00%	0.000100
6	100	15.09	-0.0000	100.00%	0.000100
9	150	22.70	-0.0000	100.00%	0.000100
12	200	30.28	-0.0000	100.00%	0.000100
14	250	37.83	3.1885	80.00%	0.000100
17	300	45.37	3.1885	80.00%	0.000100
20	350	52.95	-0.0000	100.00%	0.000100
23	400	60.52	-0.0000	100.00%	0.000100
25	450	68.10	-0.0000	100.00%	0.000100
28	500	75.68	3.1885	80.00%	0.000100
31	550	83.25	-0.0000	100.00%	0.000100
34	600	90.79	-0.0000	100.00%	0.000100
37	650	98.35	-0.0000	100.00%	0.000100
39	700	105.96	3.1885	80.00%	0.000100
42	750	113.50	3.1885	80.00%	0.000100
45	800	121.06	-0.0000	100.00%	0.000100
48	850	128.60	-0.0000	100.00%	0.000100
50	900	136.16	-0.0000	100.00%	0.000100

Gambar 4.7 Hasil proses training *fold 1*

Dalam tahap *training* ini, pada *epoch* ketiga menunjukkan ketepatan pembelajaran pada 50 iterasi memiliki akurasi mencapai 80%. Akan tetapi pada

epoch keenam hingga kedua belas mengalami peningkatan akurasi hingga 100%, dan menurun lagi menjadi 80% pada *epoch* keempat belas hingga ketujuh belas. Lalu pada *epoch* kedua puluh hingga kedua puluh lima mengalami peningkatan mencapai 100%. Pada *epoch* kedua puluh delapan akurasi menurun menjadi 80% hingga pada *epoch* ketiga puluh satu hingga ketiga puluh tujuh mengalami peningkatan mencapai 100%. Kemudian akurasi mengalami penurunan lagi menjadi 80% pada *epoch* ketiga puluh sembilan hingga keempat puluh dua, lalu meningkat lagi hingga 100% pada *epoch* keempat puluh lima hingga ke *epoch* terakhir yaitu kelima puluh. Hasil plot dari *training fold 1* digambarkan seperti pada gambar 4.8.



Gambar 4.8 Hasil plot training fold 1

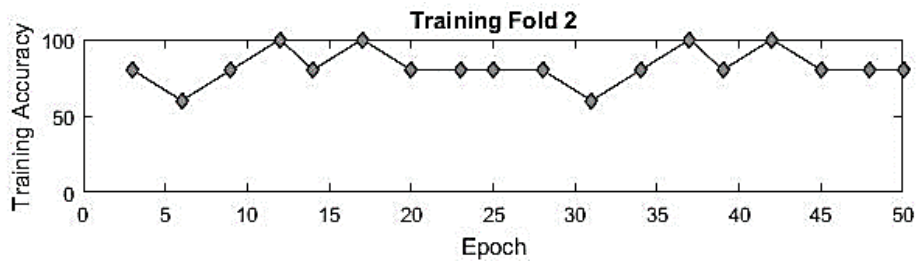
b. Hasil latih *fold 2*

Di dalam data set *fold 2*, terdapat 30 citra latih di mana terdapat 7 citra cacat dan 23 citra tidak cacat. Sementara citra sisanya yaitu 90 akan dijadikan sebagai citra latih dengan 23 citra cacat dan 67 citra tidak cacat. Hasil akurasi training dari *fold 2* dapat dilihat pada gambar 4.9.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
3	50	7.61	3.1885	80.00%	0.000100
6	100	15.21	6.3770	60.00%	0.000100
9	150	22.78	3.1885	80.00%	0.000100
12	200	30.35	-0.0000	100.00%	0.000100
14	250	37.93	3.1885	80.00%	0.000100
17	300	45.51	-0.0000	100.00%	0.000100
20	350	53.07	3.1885	80.00%	0.000100
23	400	60.64	3.1885	80.00%	0.000100
25	450	68.21	3.1885	80.00%	0.000100
28	500	75.78	3.1885	80.00%	0.000100
31	550	83.37	6.3770	60.00%	0.000100
34	600	90.93	3.1885	80.00%	0.000100
37	650	98.48	-0.0000	100.00%	0.000100
39	700	106.03	3.1885	80.00%	0.000100
42	750	113.60	-0.0000	100.00%	0.000100
45	800	121.17	3.1885	80.00%	0.000100
48	850	128.74	3.1885	80.00%	0.000100
50	900	136.31	3.1885	80.00%	0.000100

Gambar 4.9 Hasil proses training fold 2

Dalam tahap *training* ini, pada *epoch* ketiga menunjukkan ketepatan pembelajaran pada 50 iterasi memiliki akurasi mencapai 80%. Akan tetapi pada *epoch* keenam mengalami penurunan akurasi hingga 60%, dan naik lagi menjadi 80% pada *epoch* kesembilan. Dan kemudian naik lagi menjadi 100% pada *epoch* kedua belas, namun pada *epoch* keempat belas akurasi menurun menjadi 80%. Kemudian akurasi naik lagi menjadi 100% pada *epoch* ketujuh belas, namun turun lagi menjadi 80% pada *epoch* kedua puluh hingga ke *epoch* kedua puluh delapan. Akan tetapi pada *epoch* ketiga puluh satu menurun hingga menjadi 60%, lalu naik lagi pada *epoch* ketiga puluh empat menjadi 80%. Pada *epoch* ketiga puluh tujuh akurasi naik menjadi 100%, namun pada *epoch* ketiga puluh sembilan turun lagi menjadi 80%, lalu naik lagi menjadi 100% pada *epoch* keempat puluh dua. Hingga pada *epoch* keempat puluh lima hingga ke *epoch* terakhir yaitu kelima puluh mengalami penurunan akurasi mencapai 80%. Hasil plotting dari *training fold 2* digambarkan seperti pada gambar 4.10



Gambar 4.10 Hasil plot training fold 2

c. Hasil latih fold 3

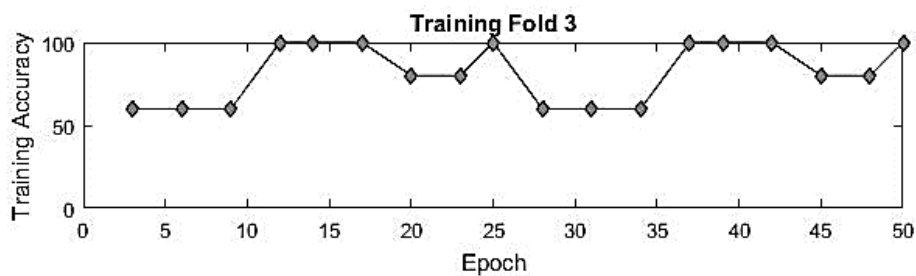
Di dalam data set *fold 3*, terdapat 30 citra latih di mana terdapat 7 citra cacat dan 23 citra tidak cacat. Sementara citra sisanya yaitu 90 akan dijadikan sebagai citra latih dengan 23 citra cacat dan 67 citra tidak cacat. Hasil akurasi training dari *fold 3* dapat dilihat pada gambar 4.11.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
3	50	7.54	6.3770	60.00%	0.000100
6	100	15.09	6.3770	60.00%	0.000100
9	150	22.65	6.3770	60.00%	0.000100
12	200	30.23	-0.0000	100.00%	0.000100
14	250	37.84	-0.0000	100.00%	0.000100
17	300	45.40	-0.0000	100.00%	0.000100
20	350	52.95	3.1885	80.00%	0.000100
23	400	60.50	3.1885	80.00%	0.000100
25	450	68.05	-0.0000	100.00%	0.000100
28	500	75.61	6.3770	60.00%	0.000100
31	550	83.14	6.3770	60.00%	0.000100
34	600	90.85	6.3770	60.00%	0.000100
37	650	98.42	-0.0000	100.00%	0.000100
39	700	105.99	-0.0000	100.00%	0.000100
42	750	113.56	-0.0000	100.00%	0.000100
45	800	121.11	3.1885	80.00%	0.000100
48	850	128.66	3.1885	80.00%	0.000100
50	900	136.22	-0.0000	100.00%	0.000100

Gambar 4.11 Hasil proses training fold 3

Dalam tahap *training* ini, pada *epoch* ketiga hingga kesembilan menunjukkan ketepatan pembelajaran pada 50 - 150 iterasi memiliki akurasi 60%. Akan tetapi pada *epoch* kedua belas hingga ketujuh belas mengalami peningkatan akurasi hingga 100%, lalu turun lagi menjadi 80% pada *epoch* kedua puluh hingga kedua

puluh tiga. Pada *epoch* kedua puluh lima akurasi meningkat menjadi 100%, namun akurasi menurun lagi menjadi 60% pada *epoch* kedua puluh delapan hingga ketiga puluh empat. Kemudian akurasi naik lagi menjadi 100% pada *epoch* ketiga puluh tujuh hingga keempat puluh dua. Pada *epoch* keempat puluh lima hingga keempat puluh delapan akurasi turun lagi menjadi 80%. Hingga mengalami peningkatan akurasi mencapai 100% pada iterasi terakhir yaitu *epoch* kelima puluh. Hasil plotting dari *training fold 3* digambarkan seperti pada gambar 4.12



Gambar 4.12 Hasil plot training fold 3

d. Hasil latih *fold 4*

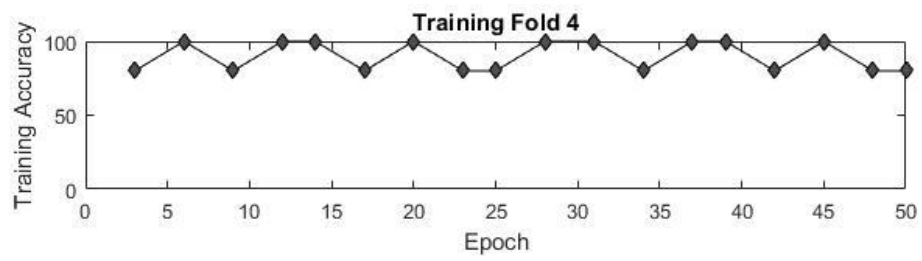
Di dalam data set *fold 4*, terdapat 30 citra latih di mana terdapat 8 citra cacat dan 22 citra tidak cacat. Sementara citra sisanya yaitu 90 akan dijadikan sebagai citra latih dengan 22 citra cacat dan 68 citra tidak cacat. Hasil akurasi training dari *fold 4* dapat dilihat pada gambar 4.13.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
3	50	7.51	1.5230	80.00%	0.000100
6	100	15.05	-0.0000	100.00%	0.000100
9	150	22.60	3.1885	80.00%	0.000100
12	200	30.15	-0.0000	100.00%	0.000100
14	250	37.72	-0.0000	100.00%	0.000100
17	300	45.32	3.1885	80.00%	0.000100
20	350	52.86	-0.0000	100.00%	0.000100
23	400	60.39	3.1885	80.00%	0.000100
25	450	67.95	3.1885	80.00%	0.000100
28	500	75.50	-0.0000	100.00%	0.000100
31	550	83.06	-0.0000	100.00%	0.000100
34	600	90.61	3.1885	80.00%	0.000100
37	650	98.16	-0.0000	100.00%	0.000100
39	700	105.72	-0.0000	100.00%	0.000100
42	750	113.27	3.1885	80.00%	0.000100
45	800	120.82	-0.0000	100.00%	0.000100
48	850	128.36	3.1885	80.00%	0.000100
50	900	135.91	3.1885	80.00%	0.000100

Gambar 4.13 Hasil proses training fold 4

Dalam tahap *training* ini, pada *epoch* ketiga menunjukkan ketepatan pembelajaran pada 50 iterasi memiliki akurasi mencapai 80%. Akan tetapi pada *epoch* keenam mengalami peningkatan akurasi hingga 100%, dan menurun lagi menjadi 80% pada *epoch* kesembilan. Lalu pada *epoch* kedua belas hingga keempat belas mengalami peningkatan mencapai 100%. Pada *epoch* ketujuh belas akurasi menurun menjadi 80% hingga pada *epoch* kedua puluh mengalami peningkatan mencapai 100%. Kemudian akurasi mengalami penurunan lagi menjadi 80% pada *epoch* kedua puluh tiga hingga kedua puluh lima, lalu meningkat lagi hingga 100% pada *epoch* kedua puluh delapan hingga ketiga puluh satu. Namun pada *epoch* ketiga puluh empat mengalami penurunan mencapai 80%, lalu naik lagi menjadi 100% pada *epoch* ketiga puluh tujuh hingga ketiga puluh sembilan. Lalu pada *epoch* keempat puluh dua menurun lagi menjadi 80% dan naik lagi menjadi 100% pada *epoch* keempat puluh lima. Hingga pada *epoch* keempat puluh delapan hingga ke *epoch* terakhir yaitu kelima puluh mengalami penurunan

akurasi menjadi 80%. Hasil plotting dari *training fold 4* digambarkan seperti pada gambar 4.14.

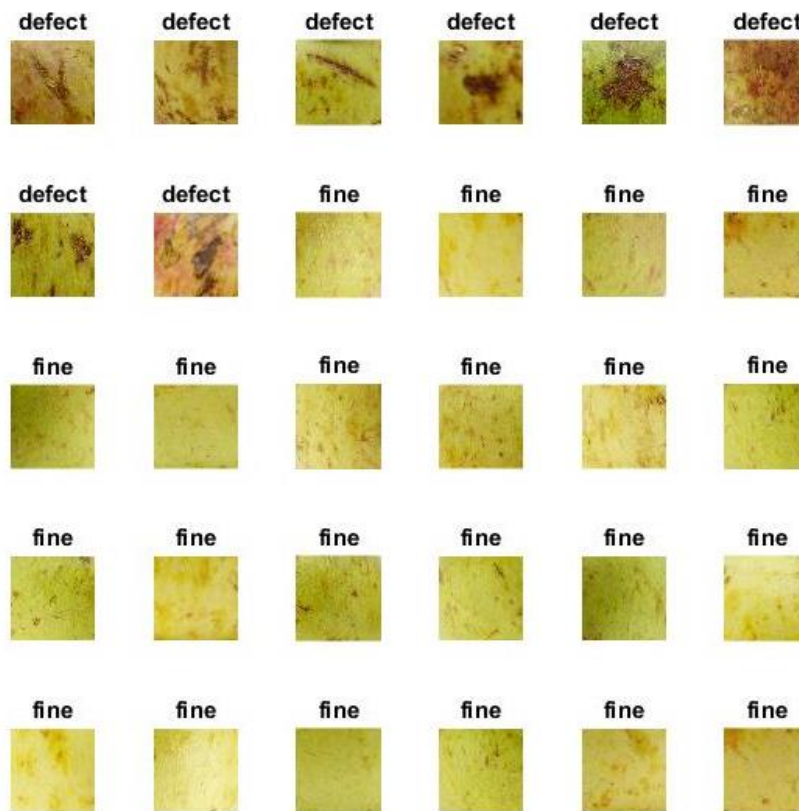


Gambar 4.14 Hasil plot training fold 4

4.3.6 Proses Uji

a. Hasil Uji *fold 1*

Dalam tahap *testing* ini, akurasi yang didapatkan mencapai akurasi penuh dengan ini tanpa kesalahan uji klasifikasi. Meskipun akurasi yang dihasilkan dari tahap *training* tidak stabil, namun program dapat menghasilkan tanpa kesalahan uji klasifikasi. Akurasi yang dihasilkan pada uji *fold 1* ini sebesar 100%. Hasil citra ditunjukkan pada gambar 4.15.

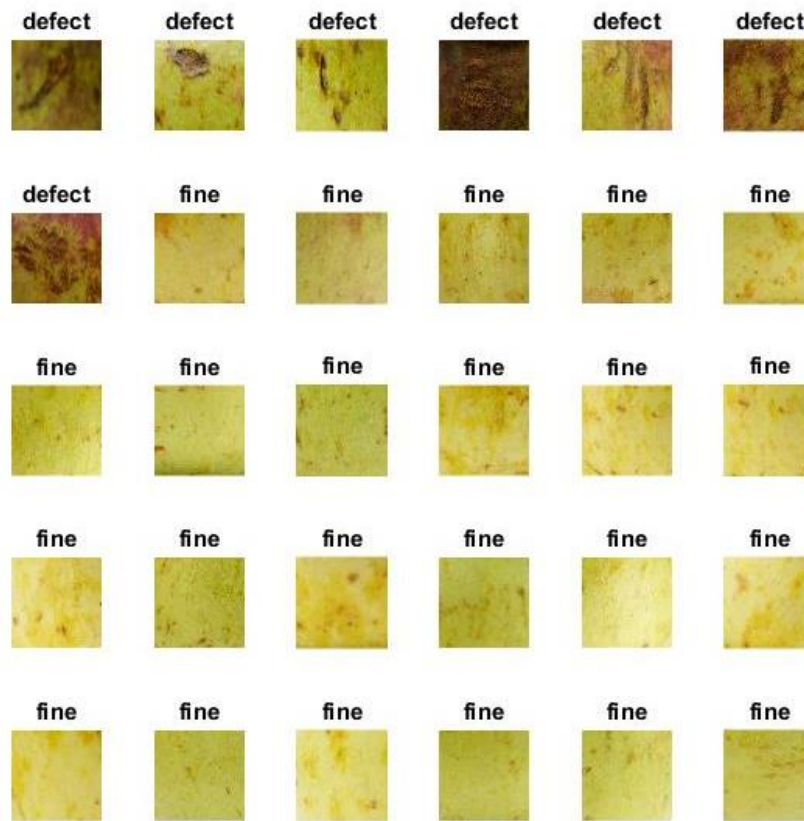


Gambar 4.15 Tidak terdapat kesalahan uji klasifikasi

b. Hasil Uji *fold 2*

Dalam tahap *testing* ini, akurasi yang didapatkan mencapai akurasi penuh dengan ini tanpa kesalahan uji klasifikasi. Meskipun akurasi yang dihasilkan dari tahap *training* tidak stabil, namun program dapat menghasilkan tanpa kesalahan uji klasifikasi.

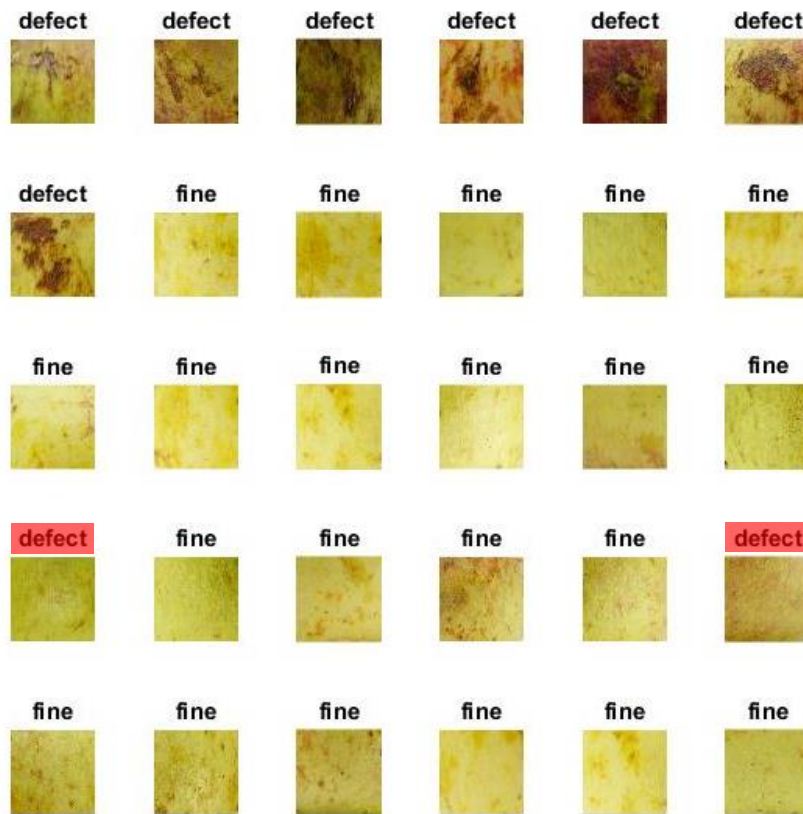
uji klasifikasi. Akurasi yang dihasilkan pada uji *fold 2* ini sebesar 100%. Hasil citra ditunjukkan pada gambar 4.16.



Gambar 4.16 Tidak terdapat kesalahan uji klasifikasi

c. Hasil Uji *fold 3*

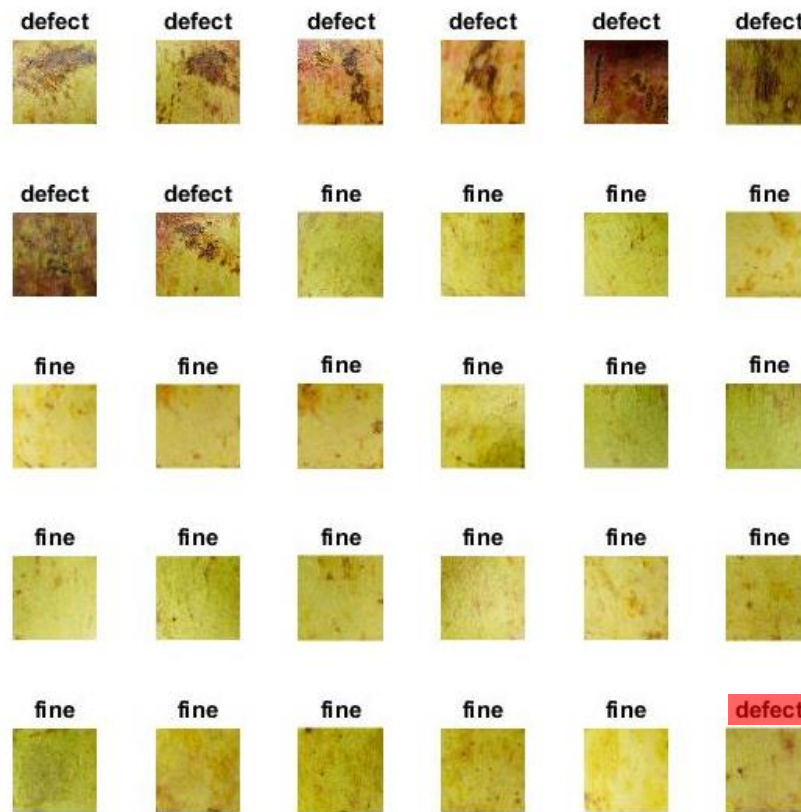
Dalam tahap *testing* ini, terdapat 2 kesalahan uji klasifikasi program. Hal ini berpengaruh dari hasil pembelajaran yang dilakukan pada tahap sebelumnya dengan akurasi yang tidak stabil. Akurasi yang dihasilkan pada uji *fold 3* ini sebesar 93,33%. Hasil citra ditunjukkan pada gambar 4.17.



Gambar 4.17 Terdapat 2 kesalahan uji klasifikasi

d. Hasil Uji *fold 4*

Dalam tahap *testing* ini, hanya terdapat 1 kesalahan uji klasifikasi program. Meskipun akurasi yang dihasilkan dari tahap *training* tidak stabil, namun program dapat menghasilkan hanya 1 kesalahan uji klasifikasi. Akurasi yang dihasilkan pada uji *fold 4* ini sebesar 96,67%. Hasil citra ditunjukkan pada gambar 4.18.



Gambar 4.18 Terdapat 1 kesalahan uji klasifikasi

4.3.6 Hasil Pengujian

Hasil Pengujian yang dilakukan pada 120 citra uji (30 citra cacat dan 90 citra tidak cacat) yang terbagi atas 4-fold data set menggunakan klasifikasi *deep learning* dengan algoritma CNN dan validasi 4-Fold Cross Validation sebagai pengklasifikasi citra. Masing-masing *fold* diuji menggunakan program deteksi yang telah dirancang, sehingga menghasilkan persentase akurasi keseluruhan dari ke-4 *fold* program yang dapat dijadikan indikator tingkat keberhasilan program. Secara lengkap presentase akurasi keseluruhan ditunjukkan pada tabel 4.6.

Tabel 4.6 Hasil Pengujian Persentase Akurasi

Algoritma	Hasil Persentase Akurasi %				Rerata
	<i>fold-1</i>	<i>fold-2</i>	<i>fold-3</i>	<i>fold-4</i>	
<i>Convolutional neural network</i>	100%	100%	93,33%	96,67%	97,5%

Kesimpulan yang didapat pada Tabel 4.6 hasil klasifikasi dan validasi 4-*fold* menghasil akurasi yang beragam, pada *fold-1* dan *fold-2* citra uji dapat diproses dengan baik oleh sistem dan menghasilkan akurasi yang tertinggi mencapai 100%. Sedangkan pada pada *fold-3* akurasi mencapai 93,33% dan pada *fold-4* menghasilkan akurasi sebesar 96,67% dan dihasilkan rerata akurasi pada ke-4 *fold* sebesar 97,5%.