

BAB IV

PEMBAHASAN DAN IMPLEMENTASI

4.1 Pendefinisian Input Dan Target

Dari data nilai RGB citra buah pepaya selanjutnya akan diolah oleh jaringan. Agar data dapat dikenali oleh jaringan, maka data harus direpresentasikan ke dalam bentuk numerik antara -1 sampai dengan 1, baik variabel maupun isinya yang merupakan masukan data nilai buah pepaya beserta kategori dan keluaran yang merupakan prediksi kematangan buah pepaya. Hal ini dikarenakan jaringan menggunakan fungsi aktivasi tansig yang rangenya dari -1 sampai 1. Nilai-nilai yang digunakan diperoleh berdasarkan kategori dari masing-masing variabel selain juga untuk memudahkan mengingat dalam pendefinisian.

4.1.1 Pendefinisian Input

Kematangan buah pepaya diubah ke dalam variabel sedangkan kategori dari masing-masing RGB tersebut adalah bentuk numerik, pendefinisian input lebih detail dapat dilihat di lampiran 1.

Tabel 4.1. Contoh Pendefinisian Input Variabel

	mean	target
R	117,698	1
G	117,698	
B	117,698	

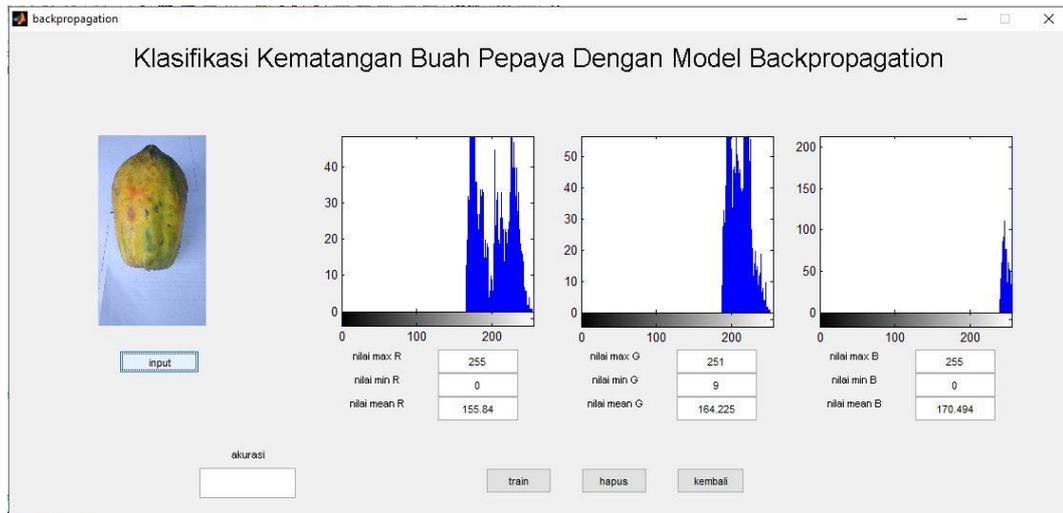
4.1.2 Pendefinisian Target

Hasil yang ingin diperoleh pada tahap ini, yaitu terdeteksinya suatu nilai untuk memprediksi kematangan buah pepaya. Hasil yang maksud adalah sebagai berikut.

- a. Jika output bernilai -1 berarti mentah.
- b. Jika output bernilai 0 berarti setengah matang.
- c. Jika output bernilai 1 berarti matang.

4.2 Hasil Data Histogram Warna

Dilakukan pemrosesan awal dengan beberapa kali pengambilan data histogram untuk buah pepaya yang memiliki tingkat kematangannya berbeda. Dibawah ini salah satu data dari beberapa pengambilan data nilai histogram warna buah pepaya dan contoh tabel data nilai histogram. tabel data nilai histogram lebih detail dapat dilihat di lampiran 2.



Gambar 4.1. Contoh Pengambilan Nilai Histogram RGB

Tabel 4.2. Contoh Data Nilai Histogram RGB Buah pepaya

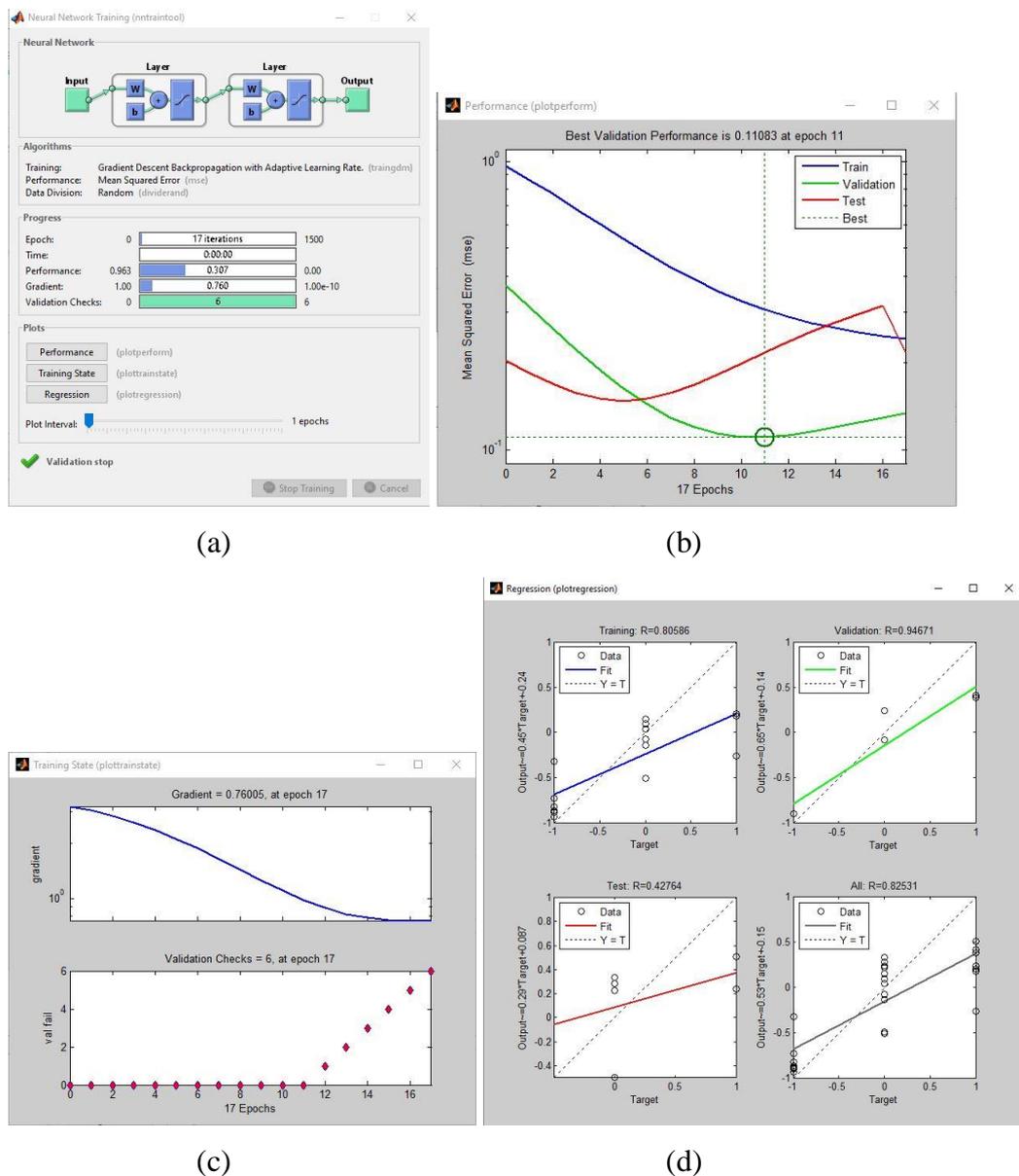
	Fitur	Mentah	Setengah matang	Matang
Mean	Red	164,386	163,92	117,698
	Green	162,646	148,465	105,374
	Blue	156,396	161,675	84,8307

4.3 Hasil Training JST

Jaringan yang telah dibangun memiliki parameter seperti pada gambar dibawah ini :

- Jumlah epoch = 1500; jumlah hidden layer = 1; performance = 0,11083.
- Garis grafik performace jaringan terjadi pada epoch ke-11.
- Nilai gradien = 0,76005, nilai cek validasi = 6 pada epoch ke-17.
- Nilai regresi untuk training = 0,80586; validation = 0,94671; test = 0,42764;

Tampilan hasil pelatihan Jaringan Syaraf Tiruan sebanyak empat kali cropping adalah sebagai berikut :

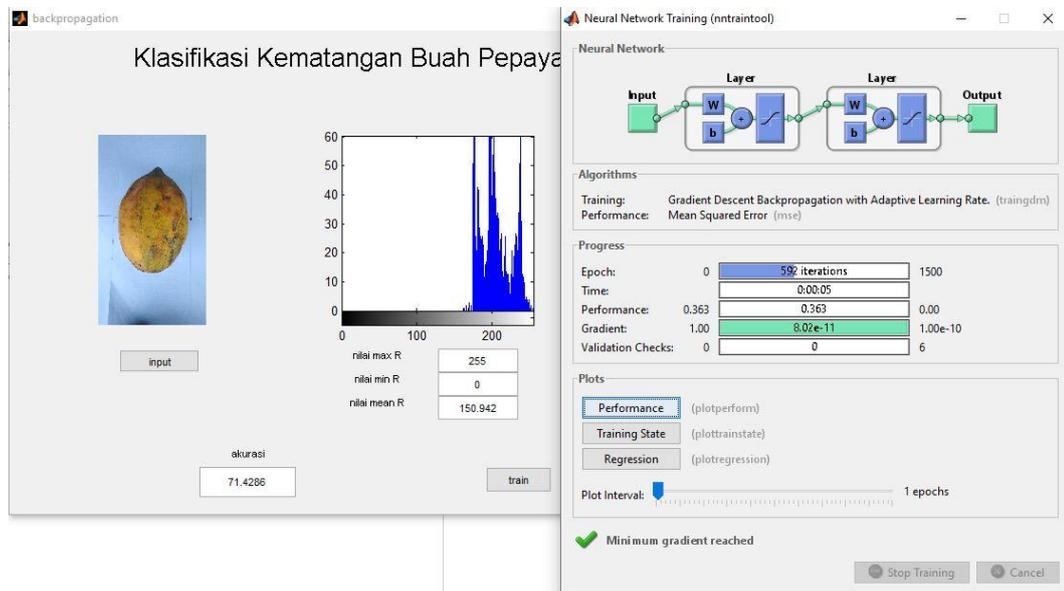


Gambar 4.2 (a) Jaringan Neural Network, (b) Grafik Mean Square Error (MSE) sampai epoch ke-11, (c) Grafik Gradient dan Validasi, (d) Grafik Regresi Training, Validasi dan Test.

4.4 Hasil Akurasi Data Training Backpropagation

Proses percobaan data training backpropagation menggunakan 3 sampel pola data yang mencirikan masing-masing tingkat kematangan buah pepaya. 1 sampel data buah pepaya mentah warna hijau, 3 buah pepaya setengah matang warna

hijau kekuningan, dan 1 sampel buah pepaya matang warna kuning, berikut citra buah pepaya yang digunakan pada proses training :



Gambar 4.3 Contoh Percobaan Training Akurasi

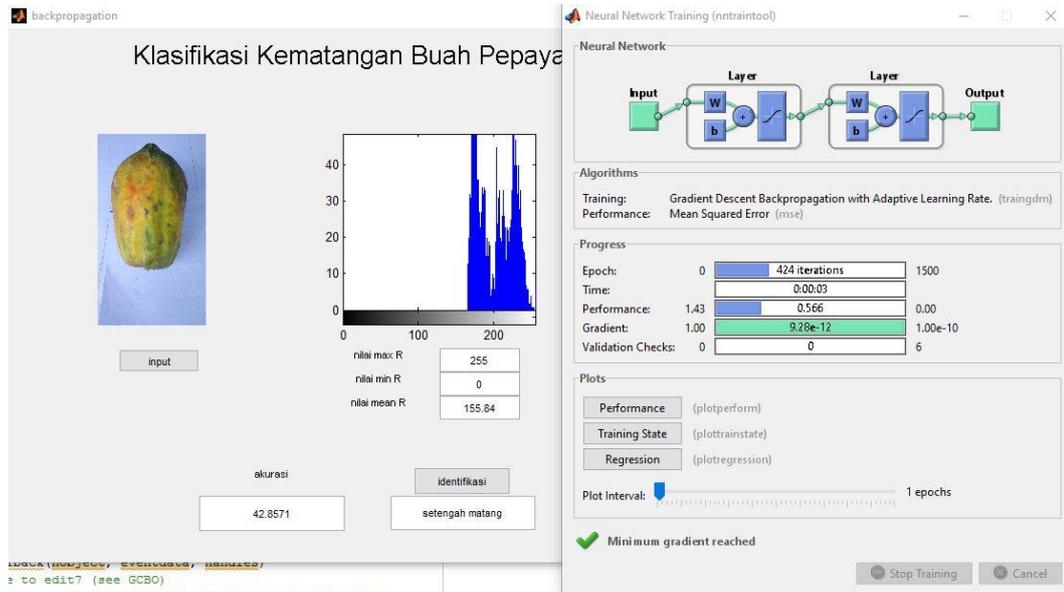
Berikut contoh tabel hasil akurasi data training. untuk lebih detail dapat dilihat di lampiran 3.

Tabel 4.3 Contoh Hasil Akurasi Data Training

No	Variabel RGB Buah Pepaya			Akurasi
	X1	X2	X3	
1	150,942	167,157	180,516	71,43%

4.5 Hasil Data Uji Backpropagation

Proses uji backpropagation menggunakan 3 sampel pola data yang mencirikan masing-masing tingkat kematangan buah pepaya. 1 sampel data buah pepaya mentah warna hijau, 6 buah pepaya setengah matang warna hijau kekuningan, dan 1 sampel buah pepaya matang warna kuning, berikut citra buah pepaya yang digunakan pada proses uji :



Gambar 4.4 Proses Uji

Berikut contoh tabel data uji variabel RGB. untuk lebih detail dapat dilihat di lampiran 4.

Tabel 4.4 Data Uji Variabel RGB

No	Variabel RGB Buah Pepaya			Target
	X1	X2	X3	
1	164,386	162,646	156,396	-1
2	151,068	146,445	140,632	0
3	163,92	148,465	161,675	0
4	147,558	127,47	129,926	0
5	139,227	119,471	122,266	0
6	157,423	133,055	136,976	0
7	159,195	134,833	140,736	0
8	117,698	105,347	84,8307	1

4.6 Pembahasan Hasil Arsitektur Backpropagation

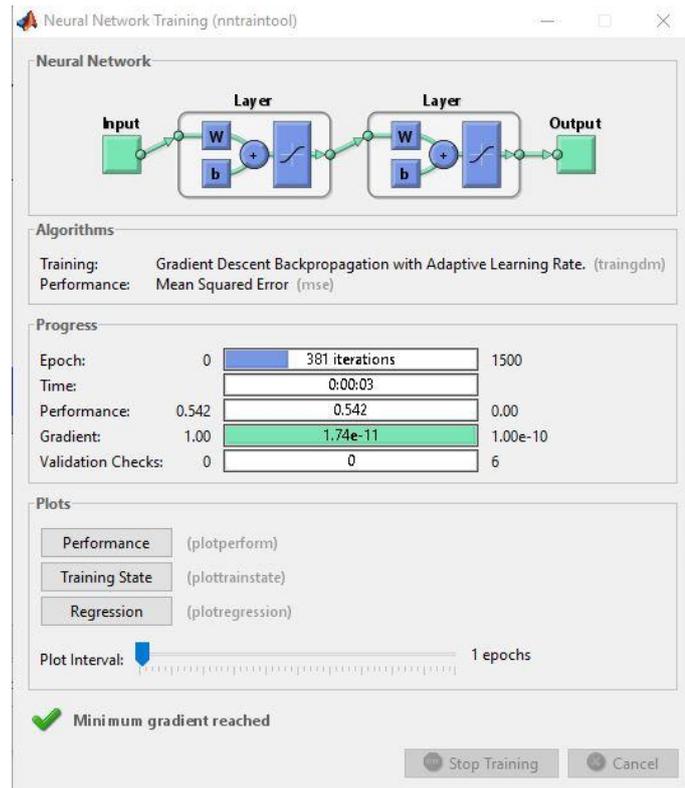
Akan dilihat pengaruh perubahan learning rate dan jumlah neuron hidden layer terhadap MSE, lama waktu pelatihan, kemudian jumlah data yang dikenali oleh sistem dengan benar.

Dari 8 data yang diujikan, diperoleh analisis sebagai berikut. Untuk masing-masing variasi, jumlah iterasi maksimum dan target error sama, yaitu:

Jumlah max iterasi (epoch) : 1500

Target error : 0

Berikut contoh proses arsitektur backpropagation disajikan dalam gambar 4.5 dan tabel hasil dari variasi arsitektur jaringan yang dilakukan.



Gambar 4.5 Contoh Proses Arsitektur Backpropagation

Berikut contoh tabel analisis hasil variasi dari arsitektur jaringan backpropagation. untuk lebih detail dapat dilihat di lampiran 5.

Tabel 4.5. Analisis Hasil Variasi Arsitektur Jaringan

No	Buah Pepaya	Epoch	MSE	Waktu (detik)	Akurasi
1		1500	0,151	0:00:28	83,33%
2		1500	0,00227	0:00:25	100%

3		1500	0,00202	0:00:26	100%
4		1500	0,00377	0:00:25	100%
5		1500	0,00235	0:00:25	100%
6		1500	0,00882	0:00:25	100%
7		1500	0,0137	0:00:25	100%
8		1500	0,00869	0:00:26	100%

4.7 Tampilan Antarmuka

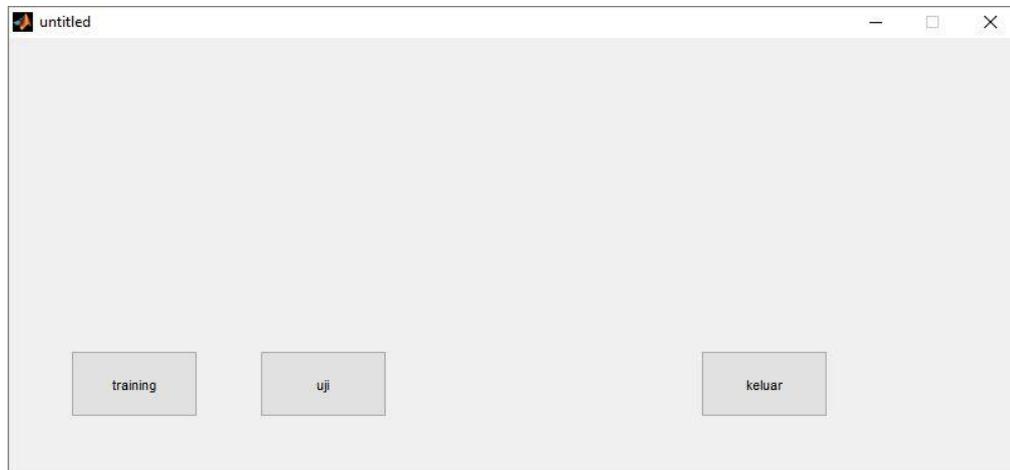
Implementasi Atarmuka adalah proses memastikan program berjalan dengan baik atau sesuai dengan harapan awal pembuat program. Sebelum dilakukan implementasi, terlebih dulu program harus dipastikan terbebas dari segala macam kesalahan. Implementasi dilaksanakan setelah perancangan atau perencanaan yang dibuat telah selesai. Implementasi program klasifikasi kematangan buah pepaya dengan model backpropagation ini terdiri dari implementasi perangkat lunak, implementasi pelatihan dan pengujian aplikasi.

Implementasi antar muka akan menampilkan implementasi tampilan dari program aplikasi dan tahap dimana aplikasi akan dipersiapkan untuk dioperasikan pada tahap selanjutnya.

Tampilan antarmuka merupakan sebuah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam

satu kesatuan yang utuh dan berfungsi. Adapun tampilan antarmuka perangkat lunak prediksi kematangan buah pepaya adalah sebagai berikut:

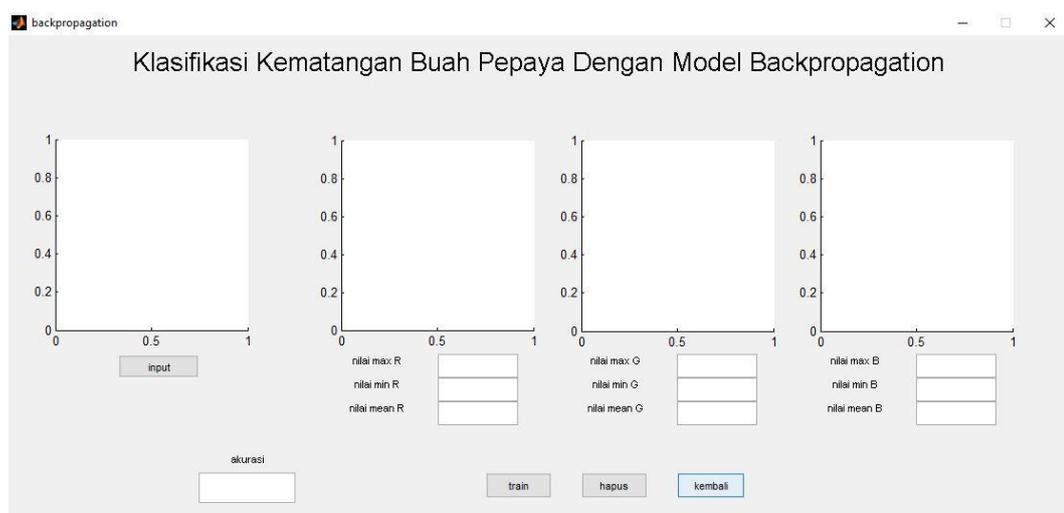
4.7.1 Tampilan Antarmuka Menu



Gambar 4.6 Tampilan Menu Klasifikasi Kematangan Buah Pepaya Model backpropagation

Tampilan menu kematangan buah pepaya model backpropagation pada Gambar 4.6 merupakan tampilan utama perangkat lunak yang memuat proses ke interface training dan uji dari sistem model backpropagation. Klik tombol “training” untuk masuk ke interface aplikasi data training citra buah pepaya. Klik tombol “uji” untuk masuk ke interface aplikasi data uji citra buah pepaya. Klik “keluar” untuk keluar dari aplikasi backpropagation.

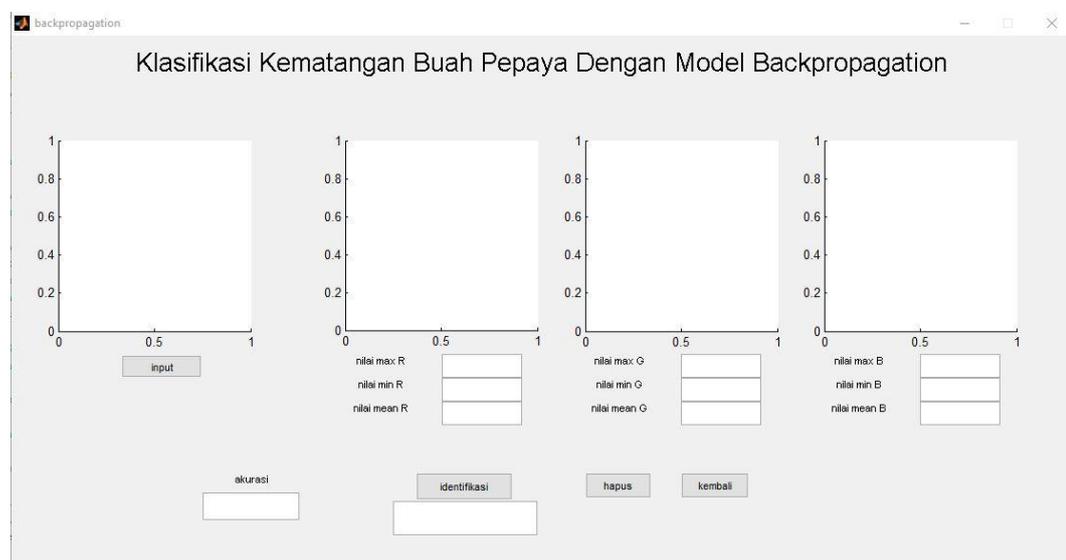
4.7.2 Tampilan Antarmuka training Model Backpropagation



Gambar 4.7 Tampilan Data Training Kematangan Buah Pepaya Model backpropagation

Tampilan data training prediksi kematangan buah pepaya pada Gambar 4.7 merupakan tampilan perangkat lunak yang memuat input dan output akurasi dari sistem model backpropagation. Klik tombol “input” untuk memasukkan citra buah pepaya dari komputer dan melihat hasil input di layer 1, kemudian secara otomatis akan memunculkan histogram masing-masing nilai rgb pada setiap textbox dan layer. Kemudian klik tombol train untuk melatih data kematangan buah pepaya berdasarkan nilai masing-masing rgb dengan metode backpropagation untuk memperoleh hasil akurasi training. Klik tombol “hapus” untuk menghapus semua input. Klik “kembali” untuk kembali ke menu awal dari program deteksi buah pepaya.

4.7.3 Tampilan Antarmuka Data Uji Model Backpropagation

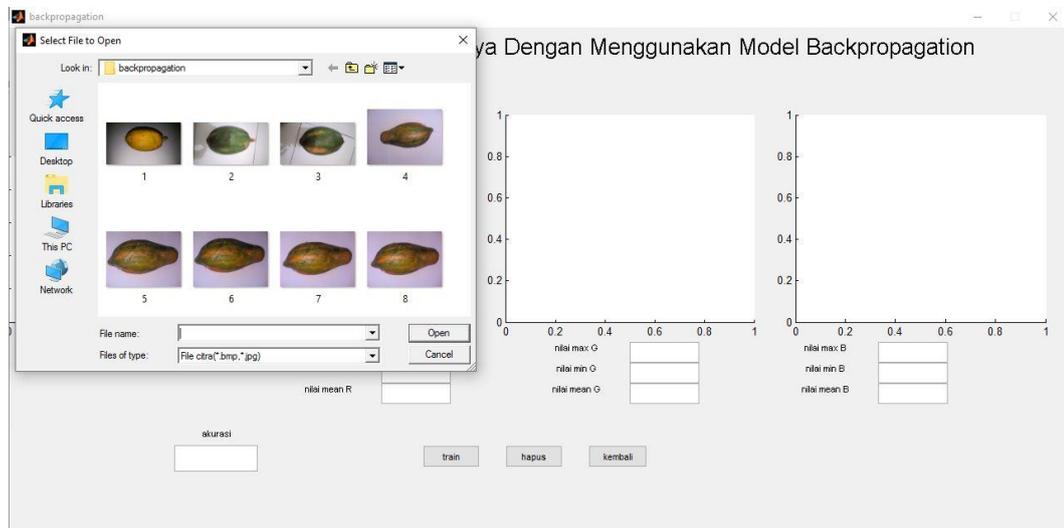


Gambar 4.8 Tampilan data uji Kematangan Buah Pepaya Model Backpropagation

Tampilan prediksi kematangan buah pepaya pada Gambar 4.8 merupakan tampilan perangkat lunak yang memuat input dan output dari sistem model backpropagation. Klik tombol “input” untuk memasukkan citra buah pepaya dari komputer dan melihat hasil input di layer 1, kemudian secara otomatis akan memunculkan histogram masing-masing nilai rgb pada setiap textbox dan layer. Kemudian klik tombol identifikasi untuk mendeteksi kematangan buah pepaya berdasarkan nilai masing-masing rgb dengan metode backpropagation. Klik tombol “hapus” untuk menghapus semua input. Klik “kembali” untuk kembali ke menu utama dari program deteksi buah pepaya.

4.8 Input citra dan Praproses Training Model Backpropagation

Proses input training citra yaitu dengan memasukkan citra ke aplikasi. Untuk input citra ke dalam aplikasi yang dilakukan adalah klik pada tombol “input”.



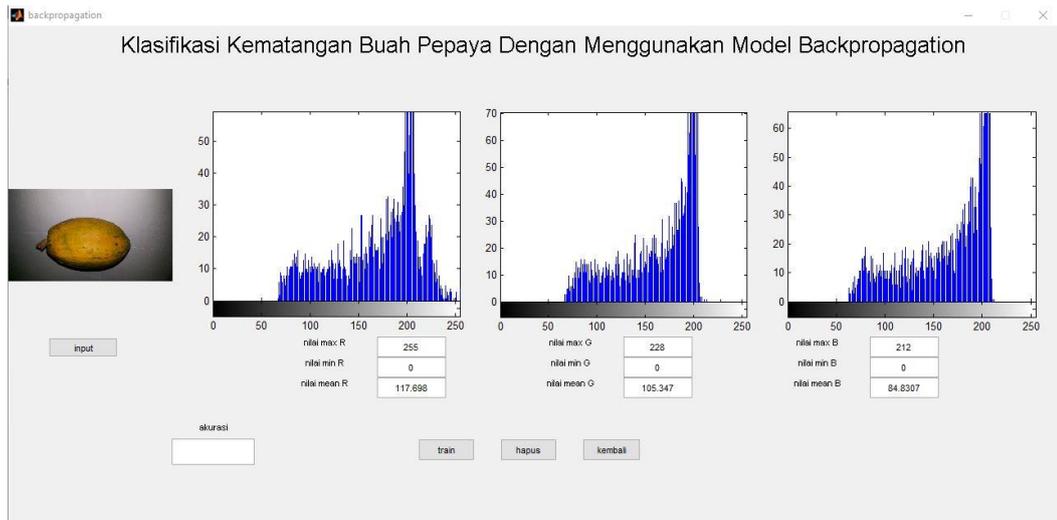
Gambar 4.9 Input training Citra

Pada Gambar 4.9 menunjukkan tampilan interface backpropagation input data training citra. Dengan menggunakan interface tombol “input” untuk input gambar ke dalam aplikasi.

4.9 Image Processing Data Training Model Backpropagation

Tahap pertama adalah melakukan proses image processing pada citra input dengan memanggil function “input”. Function input digunakan untuk Untuk input citra ke dalam aplikasi yang dilakukan adalah klik pada tombol “input” sekaligus secara otomatis akan mendeteksi nilai histogram warna image pepaya RGB berdasarkan tingkat kematangan dalam ruang warna R, G, dan B.

Hasil dari proses ini adalah input citra RGB dan hasil masing-masing komponen nilai histigram R, G, dan B yang ditampilkan dalam *ImageBox* seperti pada Gambar 4.10.

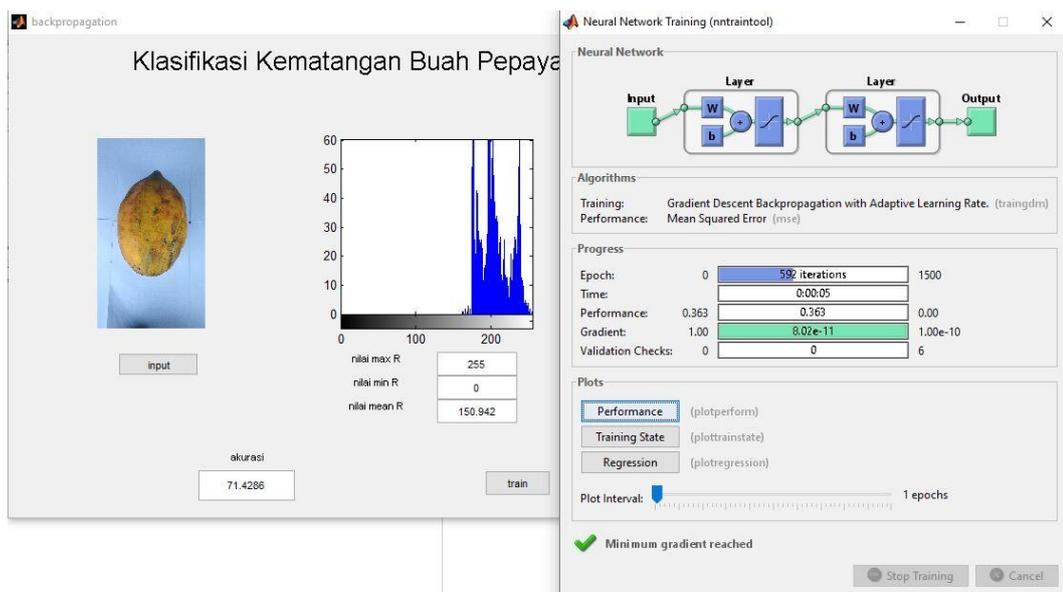


Gambar 4.10 Citra Pepaya Berdasarkan Nilai Histogram Warna R,G,B

4.10 Training Model Backpropagation

Tahap kedua adalah melakukan proses training pada citra input dengan memanggil function “train”. Function “train” digunakan untuk melatih input citra ke dalam metode backpropagation dan secara otomatis hasil akurasi akan muncul setelah data di training.

Hasil dari proses ini adalah input training citra pepaya dan hasil masing-masing komponen nilai histogram R, G, dan B yang ditampilkan seperti pada Gambar 4.11.

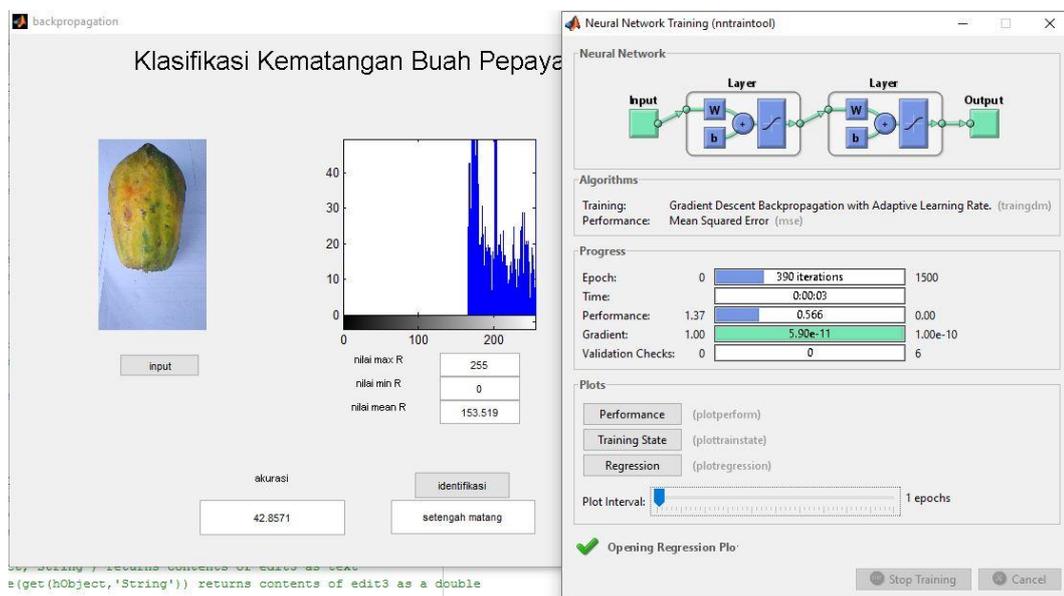


Gambar 4.11 Hasil dari proses training citra pepaya

4.11 Uji Model Backpropagation

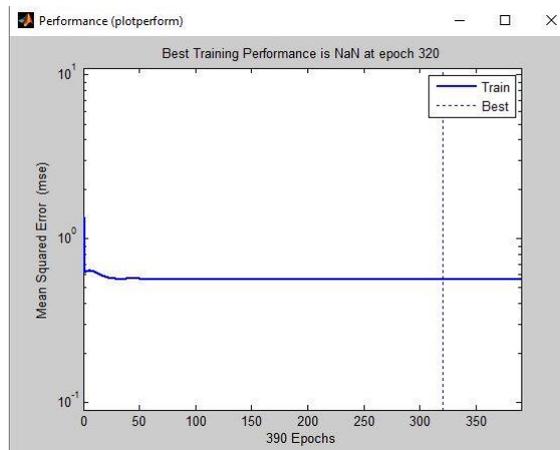
Tahap ketiga adalah melakukan proses pengujian pada citra input dengan memanggil function “identifikasi”. Function ini digunakan untuk menguji input citra ke dalam metode backpropagation dan secara otomatis hasil akurasi dan deteksi buah pepaya akan muncul setelah data di uji.

Hasil dari proses ini adalah input uji deteksi citra pepaya, hasil masing-masing komponen nilai histogram R, G, dan B dan hasil akurasi uji yang ditampilkan seperti pada Gambar 4.12.

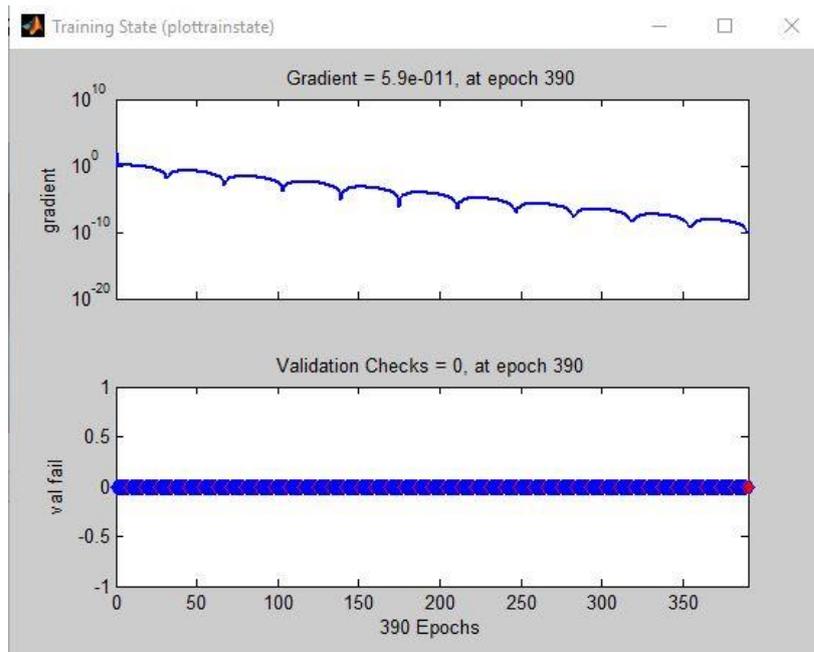


Gambar 4.12 Hasil dari proses uji citra pepaya

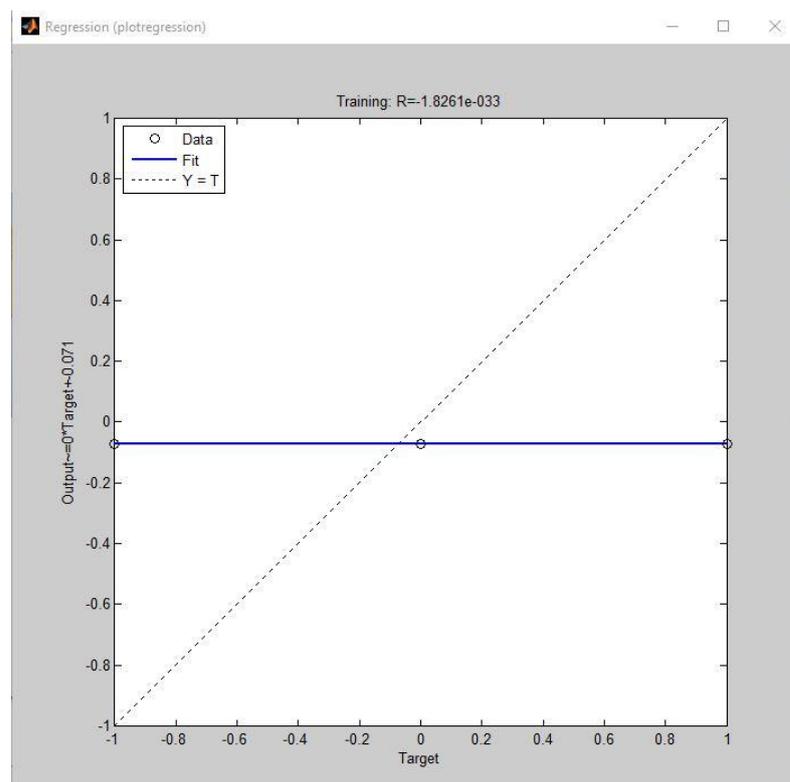
Result of Network Training



Gambar 4.13 Plot Performance



Gambar 4.14 Plot Training State



Gambar 4.15 Plot Regression

4.12 Analisis Hasil Kerja Sistem

Hasil dari pelatihan digunakan untuk menentukan konfigurasi terbaik dari jaringan dengan metode backpropagation, error goal 0, maksimum epoch 1500 serta fungsi aktivasi menggunakan fungsi tansig, sedangkan konstanta belajar dan banyaknya neuron pada lapisan tersembunyi diubah secara trial and error.

Dari hasil pelatihan yang telah dilakukan, jaringan mengenali 79,16% - 100% dari data yang dilatihkan, ditunjukkan oleh hasil keluaran yang ditampilkan sesuai dengan target yang telah ditentukan yaitu misalkan pada data ke-1 target yang diberikan adalah -1 yaitu mentah, data ke-2 dengan target 0 yaitu setengah matang, begitu juga dengan data ke-3 dengan target 1 yaitu matang.

Pada tahap pelatihan maupun pengujian sistem jaringan syaraf tiruan ini mempunyai beberapa kelebihan, antara lain proses yang akurat, cepat, serta meminimalisasi kesalahan. Pada jaringan syaraf tiruan yang perlu dilakukan adalah hanya melatih jaringan untuk “belajar” dengan cara memasukkan set data berisi sekumpulan kasus ke dalam jaringan. Dalam hal ini software yang cocok digunakan untuk membantu pemecahan masalah jaringan syaraf tiruan adalah Matlab karena banyak model jaringan syaraf tiruan yang memanipulasi matriks atau vektor dalam iterasinya. Toolbox Matlab menyediakan fungsi-fungsi khusus untuk menyelesaikan model jaringan syaraf tiruan. Adanya GUIDE Matlab yang mempunyai fungsi built-in, siap digunakan dan pemakai tidak perlu repot membuatnya sendiri sehingga lebih nyaman bila dibandingkan dengan harus menulis program seperti pada Pascal dan pemrograman bahasa lainnya.

Adapun kelemahan sistem, antara lain sistem jaringan syaraf tiruan merupakan sistem baru sehingga hanya dapat berfungsi sebagai alat bantu. Oleh sebab itu, di dalam pengambilan keputusan masih digunakan faktor-faktor pendukung atau kebijakan-kebijakan yang lain. Selain itu, belum ditemukannya cara terbaik untuk merepresentasikan data input, memilih arsitektur, serta jumlah node, dan jumlah lapisan. Cara yang digunakan hingga saat ini masih dengan cara coba-coba (trial and error).