

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Pada tahap ini akan dilakukan implementasi sistem berdasarkan hasil analisis dan perancangan yang telah dilakukan pada bab sebelumnya, hasil dari analisis dan desain yang telah dilakukan akan diimplementasikan dengan menggunakan bahasa pemrograman *Java*. Tahap implementasi sistem terdiri dari implementasi perangkat keras, implementasi perangkat lunak, dan implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Pada penelitian ini spesifikasi perangkat keras yang digunakan untuk mengimplementasikan sistem *Speaker Recognition* menggunakan algoritma *Quickprop* dapat dilihat pada Tabel 4.1 dibawah ini.

Tabel 4.1 Implementasi Perangkat Keras

No	Komponen	Spesifikasi
1	<i>Processor</i>	Intel Core I3 2.3 GHz
2	RAM	4 GB
3	<i>Harddisk</i>	500 GB

4.1.2 Implementasi Perangkat Lunak

Pada penelitian ini spesifikasi perangkat lunak yang digunakan untuk mengimplementasikan sistem *Speaker Recognition* menggunakan algoritma *Quickprop* dapat dilihat pada Tabel 4.2 dibawah ini.

Tabel 4.2 Implementasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Microsoft Windows 10 Pro
2	<i>Tools</i> Pemrograman	NetBeans IDE 8.2
		JDK 1.8

4.1.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan pada sistem *Speaker Recognition* menggunakan algoritma *Quickprop* yang dapat dilihat pada Tabel 4.3 berikut.

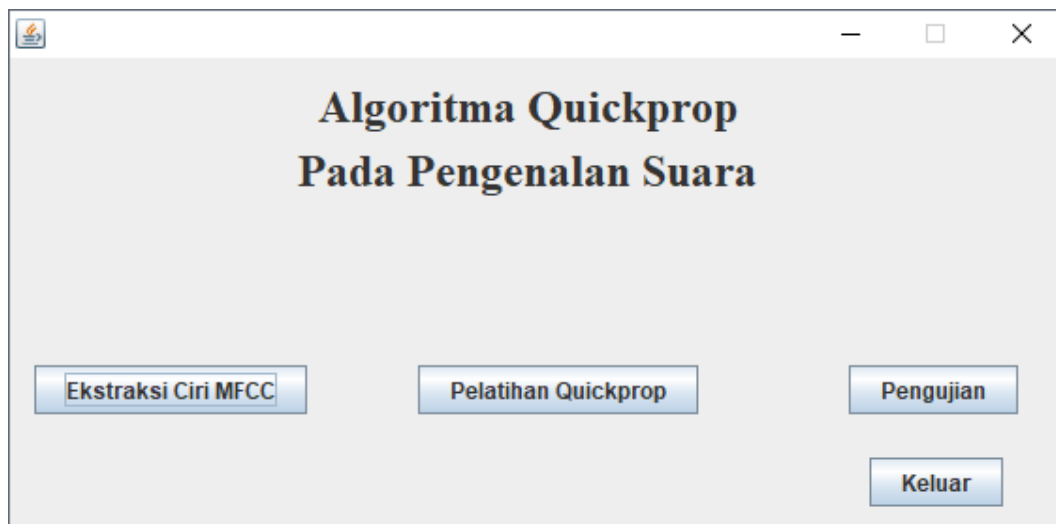
Tabel 4.3 Implementasi Antarmuka

Nama	Deskripsi	Nama File
Halaman Utama	Menampilkan halaman utama sehingga pengguna dapat memilih untuk memilih halaman atau menu yang diinginkan.	FrameUtama.java
Ekstraksi Ciri MFCC	Menampilkan halaman untuk melakukan ekstraksi ciri menggunakan MFCC terhadap file suara terpilih yang akan digunakan untuk proses pelatihan.	FrameMFCC.java
Pelatihan Quickprop	Menampilkan halaman untuk melakukan klasifikasi menggunakan algoritma <i>Quickprop</i> terhadap dataset yang telah dilakukan proses ekstraksi ciri.	FrameQuickprop.java
Pengujian Quickprop	Menampilkan halaman untuk menguji file suara terpilih dengan bobot yang telah dilakukan proses pelatihan sebelumnya.	FramePengujian.java

Berikut merupakan penjelasan antarmuka dan beberapa fungsi yang terdapat pada sistem ini.

1. Implementasi antarmuka Halaman Utama

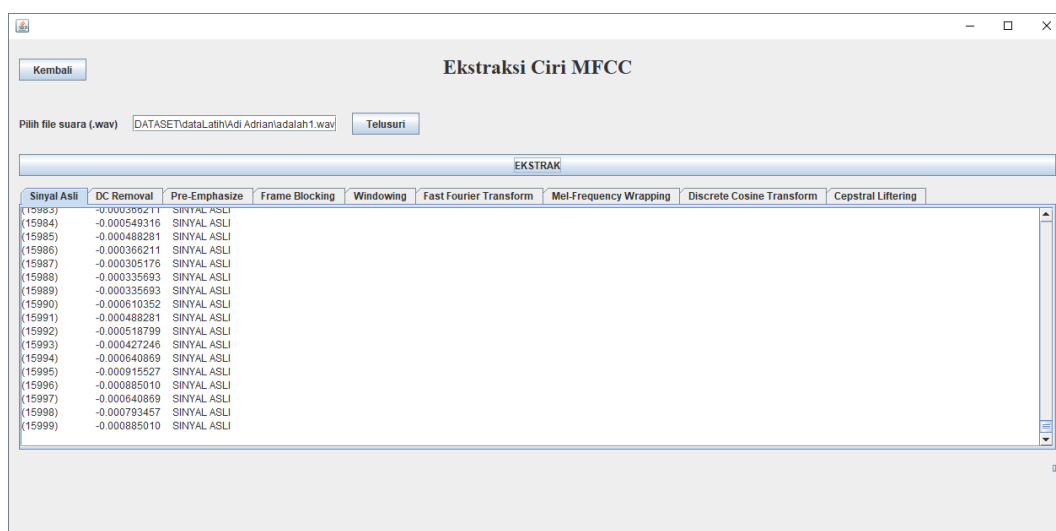
Halaman yang akan menampilkan menu-menu terkait dengan proses pengenalan suara.



Gambar 4.1 Implementasi antarmuka Halaman Utama

2. Implementasi antarmuka Halaman Ekstraksi Ciri MFCC

Halaman yang menampilkan setiap proses ekstraksi ciri file suara terpilih sehingga dapat digunakan untuk proses pembelajaran.



Gambar 4.2 Implementasi antarmuka Halaman Ekstraksi Ciri MFCC

3. Implementasi antarmuka Halaman Pelatihan Quickprop

Halaman yang menampilkan proses pelatihan yang akan dilakukan terhadap dataset dari proses ekstraksi ciri yang berguna untuk proses pengenalan selanjutnya.

Pelatihan Quickprop

Pilih file data latih: rpsiaAzid\program\Quickprop\FITUR\Fitur.txt

Data Ekstraksi Ciri	Data Target	Bobot Hasil Pelatihan				
40:-2.479489072425187	-1.4805301585664208	-2.5550771427821872	-2.471072134010094	-2.010514132087737	-2.568355130924977	-2.473
40:-3.27036928940551	-3.8370071790223608	-2.813716267578237	-2.841201546503763	-2.523456544813624	-1.9763608190659465	-1.633
40:-3.5798016586792487	-2.9205442037113474	-3.9458242062627433	-2.9815623584807045	-1.7940409666503803	-3.0734131401030607	-2.663
40:-3.321384417048849	-4.155233672747773	-3.1568443352235596	-3.9349750043621565	-3.950138232282881	-2.8342925549713702	-3.667
40:-3.8365027694205867	-2.4283075490954786	-2.740890068936198	-3.398622356616713	-4.206022204282961	-3.5087536807040105	-4.173
40:-3.903876419925631	-3.374978518778035	-3.5193652368175807	-3.2924606510922128	-3.466354148827551	-4.174099179978062	-3.721
40:-2.787890317738442	-3.0292089973264047	-3.117088371983611	-3.598540813265651	-3.539456339687265	-3.397486093128714	-3.620
40:-3.492137172780799	-3.8432596505327346	-3.9306424991976323	-6.254983149582512	-4.95157577789595	-3.2153862634113475	-3.091
40:-2.84664356501882	-2.384466248563333	-2.7448206858617628	-2.247706268511796	-2.623118582154037	-3.271655513024491	-3.119
40:-2.4716013632480456	-3.64629611715812	-1.4162291656810757	-1.0980965938542213	-3.2429353382294765	-4.03474292381898	-2.428
40:-3.2304774291489253	-3.201280609947412	-3.7275963133611536	-3.160326238367894	-2.6607250196959926	-3.24836297661563	-5.669
40:-3.172941731313291	-3.219856941871931	-2.5718932171181765	-2.1643069488962823	-2.617347878662862	-1.9107550374902047	-2.984
40:-3.4461390918554744	-3.779296763124099	-3.619476392907229	-3.845136606386809	-3.6160441912709915	-3.3256592347198812	-2.991
40:-3.4183270187477337	-2.8812416848173235	-2.467278308691524	-3.3546221055966354	-2.7375251510510563	-2.722293079502088	-3.581
40:-2.076449028963371	-2.5882379288010404	-2.7537449030302583	-2.950733358818285	-3.1490348068239946	-2.606858072574263	-2.776

Inisialisasi

Epoch:

Learning Rate:

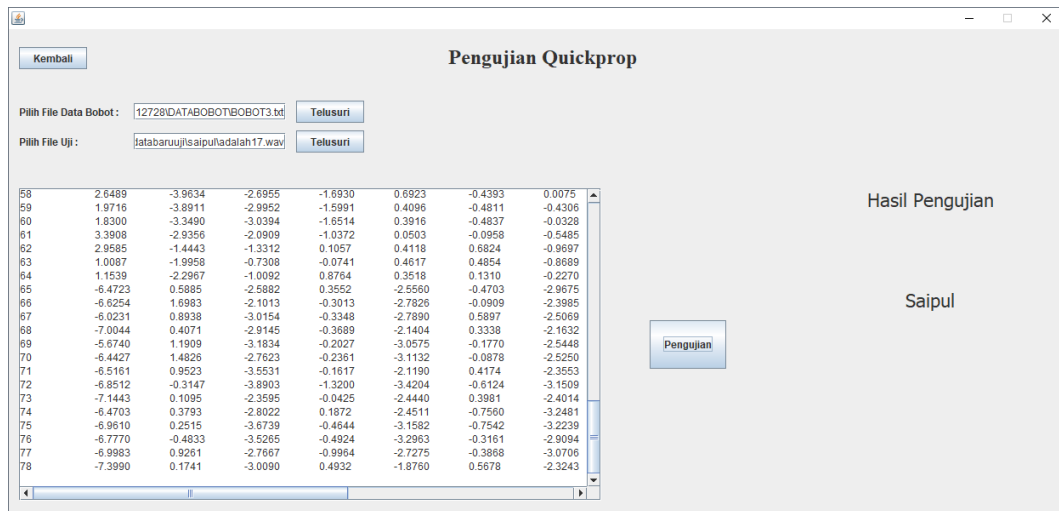
Error:

Hidden Layer:

Gambar 4.3 Implementasi antarmuka Halaman Pelatihan Quickprop

4. Implementasi antarmuka Halaman Pengujian Quickprop

Halaman yang menampilkan proses pengujian terhadap file suara terpilih lalu dilakukan proses *feedforward* terhadap bobot yang telah dilatih pada proses sebelumnya.



Gambar 4.4 Implementasi antarmuka Halaman Pengujian Quickprop

4.2 Pengujian Sistem

pengujian sistem dilakukan untuk memastikan bahwa sistem yang telah dibuat sesuai dengan perancangan sebelumnya. Pada tahap ini, pengujian sistem terdiri dari fungsionalitas dan mengukur akurasi keberhasilan yang dihasilkan oleh Algoritma *Quickprop* pada pengenalan suara.

4.2.1 Rencana Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan metode *black box* yang berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan pembangun perangkat lunak mendapat serangkaian kondisi masukan yang sepenuhnya semua persyaratan fungsional untuk suatu program. Berikut merupakan rencana pengujian *black box* dapat dilihat pada Tabel 4.4 berikut ini.

Tabel 4.4 Rencana Pengujian Fungsionalitas

No	Nama Proses	Poin Pengujian	Jenis Pengujian
1	Ekstraksi Ciri MFCC	Ekstraksi Ciri MFCC	<i>Black Box</i>
2	Pelatihan Quickprop	Pelatihan Quickprop	<i>Black Box</i>
3	Pengujian Quickprop	Pengujian Quickprop	<i>Black Box</i>

4.2.2 Hasil Pengujian Fungsionalitas

Pengujian dilakukan dengan menguji setiap fungsi untuk semua kemungkinan yang terjadi berdasarkan dari rencana pengujian yang dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Hasil Pengujian Fungsionalitas

No	Aktifitas yang dilakukan	Data Masukan	Keluaran yang Diharapkan	Pengamatan	Kesimpulan
1	Ekstraksi Ciri MFCC	File suara yang akan dilakukan ekstraksi (.wav)	File txt yang berisi hasil ekstraksi file suara terpilih	Menghasilkan file txt yang berisi nilai hasil ekstraksi ciri dengan MFCC	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
2	Pelatihan Quickprop menggunakan data latih	File txt hasil ekstraksi ciri dengan MFCC	Nilai bobot pada Input menuju Hidden, Hidden menuju Output dan Epochnya	Menampilkan nilai bobot Input menuju Hidden, Hidden menuju Output dan Epochnya	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak
3	Pengujian Quickprop berdasarkan bobot latih dan data uji	File suara (.wav) yang akan dilakukan pengujian	Nama pembicara	Menampilkan nama pembicara berdasarkan hasil pengujian	<input checked="" type="checkbox"/> Diterima <input type="checkbox"/> Ditolak

4.2.3 Skenario Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui tingkat akurasi yang dilakukan oleh sistem menggunakan algoritma *quickprop*. Awal mula data suara didapatkan dengan melakukan proses perekaman dengan mengucapkan beberapa kalimat dengan masing-masing perulangan sebanyak 10 kali, kemudian akan dipotong berdasarkan kata-kata yang berada pada depan dan tengah kalimat menjadi satu detik sehingga total data berjumlah 500 data suara. Adapun kalimat yang digunakan untuk perekaman yaitu :

Kata pada depan kalimat yang diucapkan sepuluh kali pengulangan :

1. Adalah : adalah orang itu yang mencarinya.
2. Bisa : bisa ular yang sangat mematikan.
3. Milik : milik siapakah buku itu.
4. Rupa : rupa mawar tersebut sangat indah.
5. Sebut : sebut saja dialah orangnya.

Kata pada tengah kalimat yang diucapkan sepuluh kali pengulangan :

1. Adalah : mereka itu adalah seorang mahasiswa.
2. Bisa : seorang guru harus bisa mengajar.
3. Milik : penggaris itu milik temannya.
4. Rupa : buruk rupa namun berhati baik.
5. Sebut : dia sering sebut nama kakaknya.

Berikut ini merupakan skenario pengujian yang akan dilakukan :

1. Skenario pengujian parameter

Pada skenario pengujian parameter ini akan dicari parameter yang optimal untuk digunakan sebagai proses pelatihan, dimana dataset yang digunakan adalah data latih dan data uji yang sama sebanyak 500 data. Parameter yang akan dilakukan kombinasi yaitu nilai *learning rate*, sedangkan untuk parameter lain dipilih dengan nilai yang sama, yaitu jumlah hidden layer=10, error=0.01, dan epoch=500.

2. Skenario pengujian *K-Folds Cross Validation*

Berdasarkan hasil optimal dari pengujian parameter, kemudian pengujian dilakukan menggunakan *K-Folds Cross Validation* yaitu salah satu teknik untuk

menilai/memvalidasi keakuratan sebuah model yang dibangun berdasarkan dataset tertentu. Pengujian menggunakan nilai $K=5$ yang menunjukkan jumlah dataset, dimana dataset dibagi menjadi 5 bagian yaitu K1, K2, K3, K4, K5 dengan iterasi pengujian dilakukan sebanyak 5 kali dimana tiap dataset akan secara bergantian menjadi data latih dan data uji.

Jumlah dataset yang digunakan sebanyak 500 data. Tiap dataset (K) berjumlah 100 data yang terdiri dari 5 orang pembicara mengucap 5 kata berbeda dalam bahasa indonesia (adalah, bisa, milik, rupa, sebut). Skenario pengujian *K-Folds Cross Validation* yang akan dilakukan dapat dilihat pada Tabel 4.6 dibawah ini.

Tabel 4.6 Skenario Pengujian K-Fold Cross Validation

Fold	Data Latih	Data Uji
1	K2, K3, K4, K5	K1
2	K1, K3, K4, K5	K2
3	K1, K2, K4, K5	K3
4	K1, K2, K3, K5	K4
5	K1, K2, K3, K4	K5

4.2.4 Hasil Skenario Pengujian Akurasi

Berikut ini merupakan hasil pengujian berdasarkan skenario pengujian untuk masing-masing skenario :

1. Hasil Skenario Pengujian parameter

Pada hasil pengujian ini akan dipilih nilai parameter optimal yang akan digunakan selanjutnya untuk proses pengujian *K-Folds Cross Validation*. Hasil skenario pengujian parameter dapat dilihat pada Tabel 4.7 dibawah ini.

Tabel 4.7 Kombinasi Parameter

No	<i>Learning Rate</i>	Akurasi
1	0.2	76.8
2	0.3	98.4
3	0.4	82.8
4	0.5	78.6

Berdasarkan hasil pengujian kombinasi parameter *learning rate* pada Tabel 4.7, maka akan dipilih *learning rate* = 0.3 sebagai parameter untuk pengujian *K-Fold Cross Validation* karena menghasilkan nilai akurasi tertinggi dibanding dengan jumlah yang lain.

2. Hasil skenario pengujian K-Folds Cross Validation

Berikut ini merupakan hasil pengujian skenario pengujian *K-Folds Cross Validation* untuk masing-masing nilai K :

a. Pengujian satu

Pada pengujian satu digunakan 400 data latih dan 100 data uji. Adapun data latih yang digunakan yaitu K2, K3, K4, K5 sedangkan untuk data uji yaitu K1. Hasil dari pengujian satu menggunakan *5-fold cross validation* dapat dilihat pada Tabel 4.8 dibawah ini.

Tabel 4.8 Pengujian Satu K-Fold Cross Validation

		Keluaran Sistem				
		Agung	Fadil	Herman	Jainal	Saipul
Suara Asli	Agung	18	0	0	2	0
	Fadil	0	20	0	0	0
	Herman	0	0	18	1	0
	Jainal	2	1	0	17	1
	Saipul	0	0	1	0	19

b. Pengujian Dua

Pada pengujian dua digunakan 400 data latih dan 100 data uji. Adapun data latih yang digunakan yaitu K1, K3, K4, K5 sedangkan untuk data uji yaitu K2.

Hasil dari pengujian dua menggunakan *5-fold cross validation* dapat dilihat pada Tabel 4.9 dibawah ini.

Tabel 4.9 Pengujian Dua K-Fold Cross Validation

		Keluaran Sistem				
		Agung	Fadil	Herman	Jainal	Saipul
Suara Asli	Agung	18	0	0	2	0
	Fadil	0	20	0	0	0
	Herman	0	0	19	1	0
	Jainal	1	0	1	18	0
	Saipul	0	0	0	0	20

c. Pengujian Tiga

Pada pengujian tiga digunakan 400 data latih dan 100 data uji. Adapun data latih yang digunakan yaitu K1, K2, K4, K5 sedangkan untuk data uji yaitu K3. Hasil dari pengujian tiga menggunakan *5-fold cross validation* dapat dilihat pada Tabel 4.10 dibawah ini.

Tabel 4.10 Pengujian Tiga K-Fold Cross Validation

		Keluaran Sistem				
		Agung	Fadil	Herman	Jainal	Saipul
Suara Asli	Agung	19	0	0	0	1
	Fadil	1	19	0	0	0
	Herman	0	0	20	0	0
	Jainal	0	0	2	18	0
	Saipul	0	0	1	2	17

d. Pengujian Empat

Pada pengujian empat digunakan 400 data latih dan 100 data uji. Adapun data latih yang digunakan yaitu K1, K2, K3, K5 sedangkan untuk data uji yaitu K4. Hasil dari pengujian empat menggunakan *5-fold cross validation* dapat dilihat pada Tabel 4.11 dibawah ini.

Tabel 4.11 Pengujian Empat K-Fold Cross Validation

		Keluaran Sistem				
		Agung	Fadil	Herman	Jainal	Saipul
Suara Asli	Agung	19	0	0	1	0
	Fadil	1	19	0	0	0
	Herman	0	0	20	0	0
	Jainal	0	0	1	18	1
	Saipul	0	0	1	3	16

e. Pengujian Lima

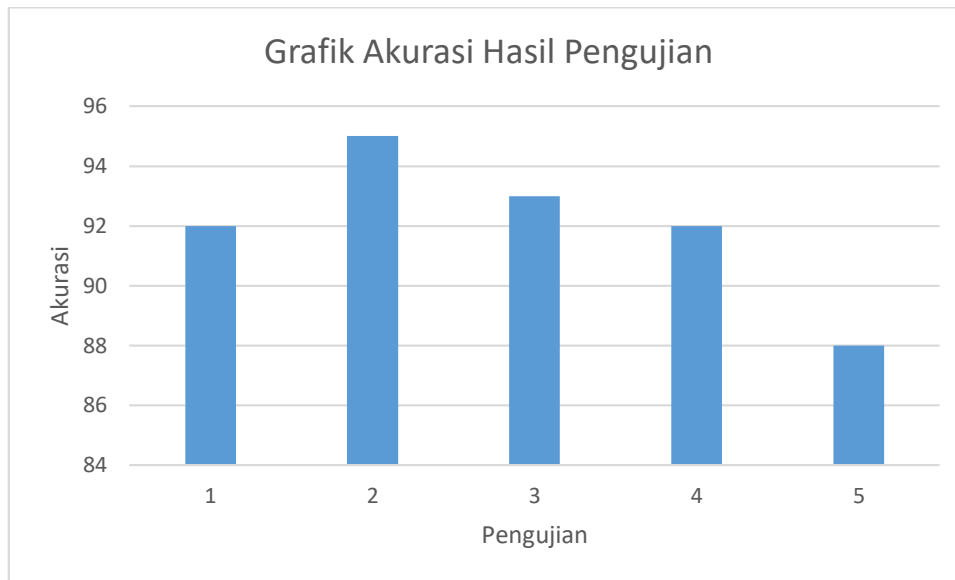
Pada pengujian lima digunakan 400 data latih dan 100 data uji. Adapun data latih yang digunakan yaitu K1, K2, K3, K4 sedangkan untuk data uji yaitu K5. Hasil dari pengujian lima menggunakan *5-fold cross validation* dapat dilihat pada Tabel 4.12 dibawah ini.

Tabel 4.12 Pengujian Lima K-Fold Cross Validation

		Keluaran Sistem				
		Agung	Fadil	Herman	Jainal	Saipul
Suara Asli	Agung	18	0	0	0	2
	Fadil	0	20	0	0	0
	Herman	0	0	18	1	1
	Jainal	0	0	2	14	4
	Saipul	0	0	1	1	18

4.2.5 Evaluasi Hasil Pengujian Akurasi

Berdasarkan semua skenario pengujian *k-fold cross validation* pada algoritma *quickprop* yang telah dilakukan dapat dilihat dalam bentuk grafik. Grafik akurasi hasil pengujian *5-fold cross validation* dapat dilihat pada Gambar 4.5 dibawah.



Gambar 4.5 Grafik Akurasi Hasil Pengujian

Berdasarkan hasil pengujian *5-Fold Cross Validation* yang terdapat pada Gambar 4.5, dengan menggunakan parameter optimal yaitu jumlah *hidden layer*=10, *learning rate*=0.3, *error*=0.01, dan maksimum *epoch*=500 telah didapatkan nilai rata-rata hasil akurasi sebesar 92%. Nilai akurasi tertinggi didapatkan sebesar 95% dan nilai akurasi terendah sebesar 88%.