BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi dan pengujian merupakan tahap lanjutan setelah perancangan sistem. Pada tahap ini sistem yang telah dibuat akan diimplementasikan pada beberapa skenario pengujian untuk melihat hasil dari sistem apakah sudah baik pada penerapan yang sebenarnya, setelah itu akan diambil kesimpulan dari hasil pengujian tersebut.

4.1.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk membangun sistem klasifikasi tingkat potensi luas lahan pertanian ini dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 Spesifikasi Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Processor Intel Core i3 2,5 GHz
2	RAM	RAM 2 GB
3	VGA	VGA 2 GB
4	Monitor	Monitor 14”
5	Harddisk yang tersedia	60 GB
6	Peranti Pendukung	Keyboard dan Mouse

4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membangun sistem klasifikasi tingkat potensi luas lahan pertanian ini dapat dilihat pada tabel 4.1 berikut.

Tabel 4.2 Spesifikasi Implementasi Perangkat Lunak

No	Keterangan	Perangkat Lunak
1	Sistem Operasi	Windows 7.0

2	Sistem Manajemen Basis Data	MySQL
3	Server Basis Data	XAMPP
4	Pengembangan Aplikasi	Microsoft Visual Studio 2010

4.1.3 Implementasi *Database*

Pembuatan *database* pada sistem klasifikasi tingkat potensi luas lahan pertanian ini dilakukan degan menggunakan DBMS MySQL. Implementasi *database* dalam bahasa SQL tersebut dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Implementasi *Database*

No	Nama Tabel	SQL
1	kecamatan	CREATE TABLE `kecamatan` (`id_kecamatan` varchar(4) NOT NULL, `nama` varchar(50) DEFAULT NULL, `lw` int(8) DEFAULT NULL, `lp` int(8) DEFAULT NULL, `plp` int(8) DEFAULT NULL, `pln` int(8) DEFAULT NULL, `tp` int(4) DEFAULT NULL, `target_output` int(2) DEFAULT NULL, PRIMARY KEY (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
2	nilai_belajar	CREATE TABLE `nilai_belajar` (`id` int(4) NOT NULL, `aLWk` decimal(20,8) DEFAULT NULL, `aLWs` decimal(20,8) DEFAULT NULL, `aLWb` decimal(20,8) DEFAULT NULL, `cLWk` decimal(20,8) DEFAULT NULL, `cLWs` decimal(20,8) DEFAULT NULL, `cLWb` decimal(20,8) DEFAULT NULL, `aLPk` decimal(20,8) DEFAULT NULL, `aLPs` decimal(20,8) DEFAULT NULL, `aLPb` decimal(20,8) DEFAULT NULL, `cLPk` decimal(20,8) DEFAULT NULL, `cLPs` decimal(20,8) DEFAULT NULL, `cLPb` decimal(20,8) DEFAULT NULL, `aPLPk` decimal(20,8) DEFAULT NULL, `aPLPs` decimal(20,8) DEFAULT NULL, `aPLPb` decimal(20,8) DEFAULT NULL, `cPLPk` decimal(20,8) DEFAULT NULL, `cPLPs` decimal(20,8) DEFAULT NULL, `cPLPb` decimal(20,8) DEFAULT NULL, `aPLNk` decimal(20,8) DEFAULT NULL,

No	Nama Tabel	SQL
		`aPLNs` decimal(20,8) DEFAULT NULL, `aPLNb` decimal(20,8) DEFAULT NULL, `cPLNk` decimal(20,8) DEFAULT NULL, `cPLNs` decimal(20,8) DEFAULT NULL, `cPLNb` decimal(20,8) DEFAULT NULL, `aTPk` decimal(20,8) DEFAULT NULL, `aTPs` decimal(20,8) DEFAULT NULL, `aTPb` decimal(20,8) DEFAULT NULL, `cTPk` decimal(20,8) DEFAULT NULL, `cTPs` decimal(20,8) DEFAULT NULL, `cTPb` decimal(20,8) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
3	temp_op_lapisan_1	CREATE TABLE `temp_op_lapisan_1` (`id` int(4) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchar(4) DEFAULT NULL, `LWk` decimal(30,8) DEFAULT NULL, `LWs` decimal(30,8) DEFAULT NULL, `LWb` decimal(30,8) DEFAULT NULL, `LPk` decimal(30,8) DEFAULT NULL, `LPs` decimal(30,8) DEFAULT NULL, `LPb` decimal(30,8) DEFAULT NULL, `PLPk` decimal(30,8) DEFAULT NULL, `PLPs` decimal(30,8) DEFAULT NULL, `PLPb` decimal(30,8) DEFAULT NULL, `PLNk` decimal(30,8) DEFAULT NULL, `PLNs` decimal(30,8) DEFAULT NULL, `PLNb` decimal(30,8) DEFAULT NULL, `TPk` decimal(30,8) DEFAULT NULL, `TPs` decimal(30,8) DEFAULT NULL, `TPb` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `temp_op_lapisan_1_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

No	Nama Tabel	SQL
4	temp_op_lapisan_2	CREATE TABLE `temp_op_lapisan_2` (`id` int(8) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchar(4) DEFAULT NULL, `w1` decimal(30,8) DEFAULT NULL, `w2` decimal(30,8) DEFAULT NULL, `w3` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `temp_op_lapisan_2_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
5	temp_op_lapisan_3	CREATE TABLE `temp_op_lapisan_3` (`id` int(8) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchar (4) DEFAULT NULL, `w^1` decimal(30,8) DEFAULT NULL, `w^2` decimal(30,8) DEFAULT NULL, `w^3` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `temp_op_lapisan_3_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
6	temp_op_lapisan_4	CREATE TABLE `temp_op_lapisan_4` (`id` int(8) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchhar (4) DEFAULT NULL, `w1f1` decimal(30,8) DEFAULT NULL, `w2f2` decimal(30,8) DEFAULT NULL, `w3f3` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `temp_op_lapisan_4_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

No	Nama Tabel	SQL
7	temp_op_lapisan_5	CREATE TABLE `temp_op_lapisan_5` (`id` int(8) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchar(4) DEFAULT NULL, `wifi_tot` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `temp_op_lapisan_5_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8	uji	CREATE TABLE `uji` (`id` int(8) NOT NULL AUTO_INCREMENT, `nama` varchar(50) DEFAULT NULL, `lw` int(8) DEFAULT NULL, `lp` int(8) DEFAULT NULL, `plp` int(8) DEFAULT NULL, `pln` int(8) DEFAULT NULL, `tp` int(4) DEFAULT NULL, `target_output` int(2) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1;
9	rmse	CREATE TABLE `rmse` (`id` int(4) NOT NULL, `rmse` double(20,8) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
10	hasil_akhir	CREATE TABLE `hasil_akhir` (`id` int(4) NOT NULL AUTO_INCREMENT, `id_kecamatan` varchar(4) DEFAULT NULL, `target_output` int(2) DEFAULT NULL, `output_jaringan` decimal(20,8) DEFAULT NULL, `error` decimal(20,8) DEFAULT NULL, PRIMARY KEY (`id`), KEY `id_kecamatan` (`id_kecamatan`), CONSTRAINT `hasil_akhir_ibfk_1` FOREIGN KEY (`id_kecamatan`) REFERENCES `kecamatan` (`id_kecamatan`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.1.4 Implementasi Antarmuka

Implementasi antarmuka yang dibuat pada sistem klasifikasi tingkat potensi luas lahan pertanian ini dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Implementasi Antarmuka

Nama	Deskripsi	Nama File
Tampilan Utama	Digunakan untuk menampilkan antarmuka utama dari sistem klasifikasi tingkat potensi luas lahan pertanian	Utama.cs
Pengolahan Data	Digunakan untuk menampilkan data masukan berupa data nama kecamatan dan parameter-parameternya. Selain itu, terdapat juga fungsi pengolahan data berupa tambah data, ubah data atau hapus data	Tampilkecamatan.cs Tambahkecamatan.cs
Pembelajaran ANFIS	Digunakan untuk melatih dan memperbarui data parameter serta melihat hasil perhitungan dari metode ANFIS	Pembelajaran.cs
Pengujian ANFIS	Digunakan untuk menghitung dan melihat hasil pengujian metode ANFIS terhadap klasifikasi tingkat potensi luas lahan pertanian berdasarkan data parameter yang sudah dilatih di pembelajaran ANFIS	Pengujian.cs

Tampilan *screenshot* dari setiap antarmuka ini dapat dilihat pada lampiran B.

4.2 Pengujian Sistem

Pengujian sistem merupakan hal penting yang bertujuan untuk menemukan kesalahan serta kekurangan yang terdapat pada sistem yang sudah dibangun. Pengujian ini bermaksud untuk mengetahui apakah perangkat lunak yang telah dibangun sudah memenuhi kriteria yang sesuai dengan tujuan perancangan. Metode yang akan digunakan pada rencana pengujian sistem ini dinamakan *blackbox*, pengujian *blackbox* merupakan metode pengujian fungsionalitas yang berfokus pada persyaratan fungsional perangkat lunaknya apakah sudah berjalan dengan sesuai. Setiap fungsi yang terdapat pada sistem akan diuji dengan dua kasus yaitu ketika menjalankannya dengan benar dan salah, untuk melihat respon dari sistem yang sudah dibangun.

4.2.1 Skenario Pengujian

Pada skenario pengujian ini, setiap poin yang diuji fungsionalitasnya akan ditampilkan untuk melihat hasil sebenarnya pada implementasi sistem klasifikasi tingkat potensi luas lahan pertanian. Setiap poin pengujian tersebut kemudian akan dijelaskan kembali lebih detail nantinya pada hasil pengujian.

Tabel 4.5 Skenario Pengujian

No	Kelas Uji	Poin Pengujian	Jenis Pengujian
1	Pengisian/Penambahan Data	Mengisi data kecamatan berupa parameter dan target outputnya	*Blackbox*
2	Pengolahan Data	- Mengubah data kecamatan - Menghapus data kecamatan	*Blackbox*
3	Pembelajaran ANFIS	- Menginputkan nilai parameter g-bell a dan c dari setiap parameter - Menginputkan kondisi pembelajaran yang akan dilakukan berdasarkan max epoh dan toleransi *error* - Menghitung nilai dari lapisan 1 sampai lapisan 5 - Menghitung nilai EBP dari lapisan 5 sampai lapisan 1 - Menghitung nilai RMSE - Menyimpan nilai parameter update	*Blackbox*
4	Pengujian ANFIS	Menampilkan hasil pengujian menggunakan data nilai belajar yang telah dilatih	*Blackbox*

4.2.2 Hasil Pengujian

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian fungsionalitas pada Implementasi ANFIS untuk klasifikasi tingkat potensi luas lahan pertanian ini sebagai berikut :

1. Pengisian/Penambahan Data

Pengisian/Penambahan data dilakukan pada menu pertama pengolahan data untuk memasukkan data kecamatan yang akan dilatih di sistem.

Tabel 4.6 Pengujian "Pengisian/Penambahan Data"

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data kecamatan yang terdiri dari nama kecamatan, lima parameter klasifikasi tingkat potensi luas lahan pertanian serta target outputnya diinputkan kedalam database sistem	Penambahan data berhasil dan disimpan kedalam database, serta muncul pesan dari sistem "Tambah Data Berhasil"	Data berhasil tersimpan dan mucul pesan "Tambah Data Berhasil"	Diterima
Proses data : Tekan tombol "Simpan"	Menyimpan data kecamatan	Penyimpanan data berhasil	Diterima
Membatalkan pengisian data : Tekan tombol "Batal"	Membatalkan penyimpanan data baru	Kembali ke menu pengolahan data	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Salah satu dari data kecamatan yang terdiri dari nama kecamatan, lima parameter klasifikasi tingkat	Penambahan data gagal dilakukan serta muncul pesan "Tambah Data Gagal"	Data gagal disimpan kedalam database dan muncul pesan "Tambah Data Gagal"	Diterima

potensi luas lahan pertanian serta target outputnya tidak diinputkan kedalam sistem			
Proses data : Tekan tombol “Simpan”	Membatalkan penyimpanan data	Penyimpanan data gagal dilakukan	Diterima

2. Pengolahan Data

Pengolahan data terdiri dari fungsi perubahan data atau penghapusan data dari data kecamatan yang ada pada data latih.

Tabel 4.7 Pengujian “Pengolahan Data”

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data yang akan diubah : Nama kecamatan “Naringgul” menjadi “Pagelaran”	Perubahan data berhasil dilakukan dan muncul pesan “Ubah Data Berhasil”	Data berhasil diubah dan mucul pesan “Ubah Data Berhasil”	Diterima
Proses data : Tekan tombol “Simpan”	Menyimpan data kecamatan yang telah diubah	Perubahan data berhasil	Diterima
Data yang akan dihapus : Nama kecamatan “Naringgul”	Penghapusan data berhasil dilakukan dan muncul pesan “Data Berhasil	Data berhasil dihapus	Diterima

	Dihapus”		
Proses data : Tekan tombol “Hapus”	Menghapus data kecamatan	Penghapusan data berhasil	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data yang akan diubah : Nama kecamatan “Naringgul” menjadi “”	Perubahan data gagal karena data yang diinputkan kosong dan muncul pesan “Ubah Data Gagal”	Data gagal diubah dan mucul pesan “Ubah Data Gagal”	Diterima
Proses data : Tekan tombol “Simpan”	Penyimpanan data kecamatan yang telah diubah gagal dilakukan	Perubahan data gagal dilakukan	Diterima
Data yang akan dihapus : Nama kecamatan “Naringgul”	Penghapusan data gagal dilakukan karena ada kesalahan pada sistem dan muncul pesan “Data Gagal Dihapus”	Data gagal dihapus	Diterima
Proses data : Tekan tombol “Hapus”	Penghapusan data kecamatan gagal dilakukan	Penghapusan data gagal dilakukan	Diterima

3. Pembelajaran ANFIS

Pembelajaran ANFIS dimaksudkan untuk melihat hasil perhitungan dari setiap langkah yang ada pada metode ANFIS yang kemudian ditampilkan oleh sistem.

Tabel 4.8 Pengujian "Pembelajaran ANFIS"

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data parameter a dan c serta kondisi pembelajaran max epoh dan toleransi error diisi pada semua field	Penghitungan ANFIS mulai berjalan kemudian setelah selesai akan menampilkan pesan "Proses Pembelajaran Selesai"	Penghitungan ANFIS berhasil dilakukan dan menampilkan pesan "Proses Pembelajaran Selesai"	Diterima
Proses data : Tekan tombol "Proses"	Mulai memproses pembelajaran ANFIS	Proses pembelajaran ANFIS selesai	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Salah satu atau banyak dari field untuk pengisian data parameter a dan c serta kondisi pembelajaran max epoh dan toleransi	Penghitungan ANFIS tidak berjalan kemudian sistem akan menampilkan pesan "Semua Field Harus Diisi"	Penghitungan ANFIS gagal dilakukan dan menampilkan pesan "Semua Field Harus Diisi"	Diterima

errornya ada yang kosong			
Proses data : Tekan tombol "Proses"	Proses pembelajaran ANFIS tidak dapat dilakukan	Proses pembelajaran ANFIS gagal	Diterima

4. Pengujian ANFIS

Pengujian ANFIS berfungi untuk melihat implementasi hasil keluaran dari sistem dalam menentukan tingkat potensi luas lahan pertanian.

Tabel 4.9 Pengujian "Pengujian ANFIS"

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Load data : data kecamatan yang akan diuji berasal dari database pengujian data	Penghitungan nilai klasifikasi dilakukan dengan menggunakan data parameter yang sudah diupdate pada proses pembelajaran	Penghitungan nilai klasifikasi berhasil dilakukan dengan menggunakan data parameter yang sudah diupdate	Diterima
Proses data : tekan tombol "Uji Coba"	Setiap langkah perhitungan ANFIS dari lapisan 1 sampai lapisan 5 berhasil dilakukan dan menampilkan	Proses eksekusi perhitungan berhasil dilakukan dan menampilkan tingkat akurasi dan RMSE	Diterima

	tingkat keakurasian serta nilai RMSE nya	pengujian	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Hasil Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Load data : terdapat kesalahan pada data kecamatan yang akan diuji seperti kekosongan data atau kesalahan tipe data	Proses perhitungan ANFIS tidak dapat dilakukan karena ada kesalahan pada database pengujiannya	Proses eksekusi perhitungan gagal dilakukan karena terdapat kesalahan pada database pengujian	Diterima
Proses data : tekan tombol “Uji Coba”	Proses perhitungan ANFIS gagal dilakukan dan nilai keakurasian serta RMSE yang ditampilkannya bernilai nol	Proses eksekusi perhitungan gagal dilakukan dan tidak dapat menampilkan nilai tingkat akurasi serta RMSE pengujiannya	Diterima

4.2.3 Kesimpulan Pengujian Sistem

Hasil dari pengujian fungsionalitas dengan metode *blackbox* yang telah dilakukan menunjukan bahwa dari sistem yang dibangun sudah memenuhi persyaratan secara fungsional, akan tetapi pada prosesnya masih memungkinkan terjadi kesalahan sehingga membutuhkan proses evaluasi untuk lebih mengetahui kekurangan dari aplikasi.

4.3 Pengujian Metode

Pengujian metode dilakukan dengan tujuan untuk mengetahui hasil performansi dari penerapan metode *Adaptive Neuro Fuzzy Inference System* (ANFIS) dalam menentukan klasifikasi tingkat potensi luas lahan pertanian. Pada pengujian metode ini akan melibatkan sejumlah 79 data kecamatan yang akan dibagi menjadi dua bagian, yaitu sebagian untuk digunakan sebagai data pembelajaran dan sebagian lainnya akan digunakan sebagai data pengujian. Persentase pembagian data kecamatan yang akan digunakan yaitu sebesar 64,5 % atau sejumlah 51 data untuk pembelajaran ANFIS, dan 35,5 % atau sejumlah 28 data untuk pengujian metode nya.

Pada pengujian metode ANFIS ini terdapat dua buah variable yang dapat digunakan untuk melihat perbedaan hasil performansi apabila diubah nilainya yaitu banyak iterasi atau epoh dan toleransi *error*. Perubahan banyak iterasi dan toleransi *error* ini juga bertujuan untuk memberikan batasan pada proses pembelajaran metode ANFIS. Hasil dari pembelajaran ANFIS yang berupa update data parameter gbell tersebut yang kemudian akan digunakan pada proses pengujian metode untuk melihat tingkat keakurasian dan *error rate* yang didapat.

Tingkat akurasi dari metode ANFIS ini dapat dilihat dari perbandingan banyaknya hasil prediksi sistem yang sesuai dan hasil prediksi sistem yang tidak sesuai dengan nilai target output yang sebenarnya. Perhitungan akurasi akan didapatkan berdasarkan hasil prediksi yang sesuai dengan nilai aktualnya dibagi dengan banyaknya data uji untuk kemudian hasilnya akan diubah menjadi nilai persentase untuk lebih memudahkan dalam mengidentifikasi nilai keakurasian yang didapat. Untuk menunjang perhitungan akurasi tersebut, maka akan digunakan metode *Confusion Matrix Multi-Class* dengan menggunakan persamaan 2.17. Selain untuk mengetahui tingkat keakurasian, penghitungan metode *Confusion Matrix Multi-Class* juga dapat digunakan untuk menghitung nilai *precision* dan *recall.* Akan tetapi, fokus utama untuk mengukur kinerja dari sistem ini tetap pada hasil dari penghitungan tingkat akurasinya, karena *precision* dan *recall* hanya digunakan untuk menghitung nilai dari lingkup suatu kategori atau kelas saja.

4.3.1 Skenario Pengujian 1

Skenario pengujian 1 dilakukan dengan menggunakan 28 data kecamatan uji coba, dimana terdapat 7 kecamatan dengan tingkat potensi luas lahan berkelanjutan, 10 kecamatan dengan tingkat potensi luas lahan cukup berkelanjutan dan 11 kecamatan dengan tingkat potensi luas lahan kurang berkelanjutan. Pada skenario ini akan digunakan toleransi *error* dari nilai 0.0020, 0.0018, 0.0016, 0.0014, 0.0012, 0.0010 dengan maksimal iterasi sebesar 600.

1. Toleransi *Error* 0.0020

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0020 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.10 berikut.

Tabel 4.10 Hasil Pengujian Toleransi *Error* 0.0020

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	3	-2
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	3	-2
7	1	3	-2
8	2	2	0
9	2	1	1
10	2	3	-1
11	2	1	1
12	2	2	0
13	2	1	1
14	2	2	0
15	2	3	-1
16	2	2	0
17	2	1	1
18	3	3	0

19	3	2	1
20	3	1	2
21	3	1	2
22	3	3	0
23	3	1	2
24	3	3	0
25	3	2	1
26	3	1	2
27	3	2	1
28	3	1	2

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.11 berikut.

Tabel 4.11 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0020

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	3	4	5	12
	2	1	4	3	8
	3	3	2	3	8
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.12.

Tabel 4.12 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0020

	Kategori		
	1	**2**	**3**
Precision	0,25	0,5	0,38

Recall	0,43	0,4	0,27
Accuracy	**0,3571**		

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0020 yaitu sebesar 35,71 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

2. Toleransi *Error* 0.0018

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0018 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.13 berikut.

Tabel 4.13 Hasil Pengujian Toleransi *Error* 0.0018

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	3	-2
2	1	2	-1
3	1	1	0
4	1	1	0
5	1	3	-2
6	1	2	-1
7	1	1	0
8	2	2	0
9	2	1	1
10	2	2	0
11	2	1	1
12	2	3	-1
13	2	2	0
14	2	2	0
15	2	3	-1
16	2	2	0
17	2	3	-1
18	3	3	0
19	3	2	1

20	3	1	2
21	3	1	2
22	3	3	0
23	3	1	2
24	3	3	0
25	3	2	1
26	3	3	0
27	3	2	1
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.14 berikut.

Tabel 4.14 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0018

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	3	2	3	8
	2	2	5	3	10
	3	2	3	5	10
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.15.

Tabel 4.15 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0018

	Kategori		
	1	**2**	**3**
Precision	0,38	0,5	0,5
Recall	0,43	0,5	0,45

Accuracy	**0,4643**

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0018 yaitu sebesar 46,43 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

3. Toleransi *Error* 0.0016

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0016 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.16 berikut.

Tabel 4.16 Hasil Pengujian Toleransi *Error* 0.0016

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	3	-2
2	1	1	0
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	2	-1
7	1	1	0
8	2	2	0
9	2	2	0
10	2	1	1
11	2	2	0
12	2	3	-1
13	2	2	0
14	2	1	1
15	2	2	0
16	2	2	0
17	2	3	-1
18	3	3	0
19	3	3	0
20	3	1	2

21	3	3	0
22	3	3	0
23	3	1	2
24	3	3	0
25	3	2	1
26	3	3	0
27	3	3	0
28	3	1	2

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.17 berikut.

Tabel 4.17 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0016

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	4	2	3	9
	2	2	6	1	9
	3	1	2	7	10
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.18.

Tabel 4.18 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0016

	Kategori		
	1	**2**	**3**
Precision	0,44	0,67	0,7
Recall	0,57	0,6	0,64
Accuracy	**0,6071**		

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0016 yaitu sebesar 60,71 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

4. Toleransi *Error* 0.0014

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0014 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.19 berikut.

Tabel 4.19 Hasil Pengujian Toleransi *Error* 0.0014

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	3	-2
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	1	0
7	1	1	0
8	2	2	0
9	2	2	0
10	2	2	0
11	2	2	0
12	2	2	0
13	2	3	-1
14	2	1	1
15	2	2	0
16	2	2	0
17	2	2	0
18	3	3	0
19	3	3	0
20	3	1	2
21	3	3	0
22	3	3	0

23	3	3	0
24	3	3	0
25	3	2	1
26	3	3	0
27	3	1	2
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.20 berikut.

Tabel 4.20 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0014

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	5	1	2	8
	2	1	8	1	10
	3	1	1	8	10
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.21.

Tabel 4.21 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0014

	Kategori		
	1	**2**	**3**
Precision	0,63	0,8	0,8
Recall	0,71	0,8	0,73
Accuracy	**0,75**		

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0014 yaitu sebesar 75 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

5. Toleransi *Error* 0.0012

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0012 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.22 berikut.

Tabel 4.22 Hasil Pengujian Toleransi *Error* 0.0012

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	2	-1
7	1	1	0
8	2	2	0
9	2	2	0
10	2	2	0
11	2	1	1
12	2	2	0
13	2	2	0
14	2	3	-1
15	2	2	0
16	2	2	0
17	2	2	0
18	3	3	0
19	3	3	0
20	3	3	0
21	3	3	0
22	3	3	0

23	3	3	0
24	3	3	0
25	3	1	2
26	3	3	0
27	3	3	0
28	3	2	1

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.23 berikut.

Tabel 4.23 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0012

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	6	1	1	8
	2	1	8	1	10
	3	0	1	9	10
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.24.

Tabel 4.24 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0012

	Kategori		
	1	**2**	**3**
Precision	0,75	0,8	0,9
Recall	0,86	0,8	0,82
Accuracy	**0,8214**		

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0012 yaitu sebesar 82,14 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

6. Toleransi *Error* 0.0010

Hasil dari skenario pengujian 1 dengan toleransi *error* 0.0010 menggunakan 28 data kecamatan dan maksimal iterasi sebanyak 600 dapat dilihat pada tabel 4.25 berikut.

Tabel 4.25 Hasil Pengujian Toleransi *Error* 0.0010

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	2	-1
7	1	1	0
8	2	2	0
9	2	2	0
10	2	2	0
11	2	2	0
12	2	2	0
13	2	2	0
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	3	3	0
19	3	3	0
20	3	3	0
21	3	3	0
22	3	3	0

23	3	3	0
24	3	3	0
25	3	2	1
26	3	3	0
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.26 berikut.

Tabel 4.26 *Confusion Matrix Multi-Class* Dari Toleransi *Error* 0.0010

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	6	0	0	6
	2	1	10	1	12
	3	0	0	10	10
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.27.

Tabel 4.27 *Precision, Recall* dan *Accuracy* Dari Toleransi *Error* 0.0010

	Kategori		
	1	**2**	**3**
Precision	1	0,83	1
Recall	0,86	1	0,91
Accuracy	**0,9286**		

Tingkat keakurasian dari hasil pengujian menggunakan toleransi *error* 0.0010 yaitu sebesar 92,86 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

7. Kesimpulan Skenario Pengujian 1

Dari hasil skenario pengujian 1 yang ditampilkan pada tabel 4.10 sampai tabel 4.27, didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.28 berikut.

Tabel 4.28 Hasil Skenario Pengujian 1

Toleransi *Error* Pembelajaran	Akurasi Pengujian	Iterasi Terakhir Pembelajaran	RMSE Pengujian
0.0020	35,71 %	82	0.832278
0.0018	46,43 %	173	0.817765
0.0016	60,71 %	259	0.723406
0.0014	75 %	325	0.612921
0.0012	82,14 %	471	0.473228
0.0010	92,86 %	516	0.247634

Dari data hasil skenario pengujian diatas dapat disimpulkan bahwa semakin kecil nilai toleransi *error* yang diberikan maka tingkat keakurasian yang didapat mejadi lebih tinggi, akan tetapi jumlah iterasi yang diperlukan pun menjadi semakin banyak.

4.3.2 Skenario Pengujian 2

Skenario pengujian 2 dilakukan dengan menggunakan 28 data kecamatan uji coba, dimana terdapat 7 kecamatan dengan tingkat potensi luas lahan berkelanjutan, 10 kecamatan dengan tingkat potensi luas lahan cukup berkelanjutan dan 11 kecamatan dengan tingkat potensi luas lahan kurang berkelanjutan. Pada skenario ini akan digunakan maksimal iterasi dari nilai 100, 200, 300, 400, 500, 600 dengan nilai toleransi *error* sebesar 0.00001

1. Maksimal Iterasi 100

Hasil dari skenario pengujian 2 dengan maksimal iterasi 100 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.29 berikut.

Tabel 4.29 Hasil Pengujian Maksimal Iterasi 100

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	2	-1
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	2	-1
7	1	2	-1
8	2	1	1
9	2	2	0
10	2	1	1
11	2	1	1
12	2	2	0
13	2	3	-1
14	2	2	0
15	2	1	1
16	2	2	0
17	2	3	-1
18	3	2	1
19	3	1	2
20	3	3	0
21	3	1	2
22	3	3	0
23	3	2	1
24	3	3	0
25	3	2	1

26	3	3	0
27	3	2	1
28	3	1	2

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.30 berikut.

Tabel 4.30 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 100

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	3	4	3	10
	2	4	4	4	12
	3	0	2	4	6
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.31.

Tabel 4.31 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 100

	Kategori		
	1	**2**	**3**
Precision	0,3	0,33	0,67
Recall	0,43	0,4	0,36
Accuracy	**0,3929**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 100 yaitu sebesar 39,29 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

2. Maksimal Iterasi 200

Hasil dari skenario pengujian 2 dengan maksimal iterasi 200 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.32 berikut.

Tabel 4.32 Hasil Pengujian Maksimal Iterasi 200

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	3	-2
2	1	1	0
3	1	2	-1
4	1	1	0
5	1	2	-1
6	1	1	0
7	1	3	-2
8	2	3	-1
9	2	2	0
10	2	2	0
11	2	2	0
12	2	1	1
13	2	2	0
14	2	3	-1
15	2	3	-1
16	2	1	1
17	2	2	0
18	3	2	1
19	3	3	0
20	3	3	0
21	3	1	2
22	3	3	0
23	3	3	0
24	3	2	1
25	3	2	1

26	3	3	0
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.33 berikut.

Tabel 4.33 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 200

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	3	2	3	8
	2	2	5	1	8
	3	2	3	7	12
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.34.

Tabel 4.34 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 200

	Kategori		
	1	**2**	**3**
Precision	0,38	0,63	0,58
Recall	0,43	0,5	0,64
Accuracy	**0,5357**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 200 yaitu sebesar 53,57 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

3. Maksimal Iterasi 300

Hasil dari skenario pengujian 2 dengan maksimal iterasi 300 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.35 berikut.

Tabel 4.35 Hasil Pengujian Maksimal Iterasi 300

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	3	-2
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	2	-1
7	1	1	0
8	2	2	0
9	2	2	0
10	2	2	0
11	2	3	-1
12	2	2	0
13	2	2	0
14	2	2	0
15	2	1	1
16	2	3	-1
17	2	2	0
18	3	2	1
19	3	3	0
20	3	3	0
21	3	3	0
22	3	2	1
23	3	3	0
24	3	3	0
25	3	1	2

26	3	3	0
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.36 berikut.

Tabel 4.36 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 300

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	4	1	1	6
	2	2	7	2	11
	3	1	2	8	11
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.37.

Tabel 4.37 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 300

	Kategori		
	1	**2**	**3**
Precision	0,67	0,64	0,73
Recall	0,57	0,7	0,73
Accuracy	**0,6786**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 300 yaitu sebesar 67,86 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

4. Maksimal Iterasi 400

Hasil dari skenario pengujian 2 dengan maksimal iterasi 400 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.38 berikut.

Tabel 4.38 Hasil Pengujian Maksimal Iterasi 400

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	2	-1
5	1	1	0
6	1	1	0
7	1	3	-2
8	2	2	0
9	2	2	0
10	2	2	0
11	2	3	-1
12	2	2	0
13	2	2	0
14	2	2	0
15	2	1	1
16	2	2	0
17	2	2	0
18	3	3	0
19	3	3	0
20	3	3	0
21	3	3	0
22	3	3	0
23	3	1	2
24	3	3	0
25	3	3	0

26	3	1	2
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.39 berikut.

Tabel 4.39 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 400

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	5	1	2	8
	2	1	8	0	9
	3	1	1	9	11
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.40.

Tabel 4.40 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 400

	Kategori		
	1	**2**	**3**
Precision	0,63	0,89	0,82
Recall	0,71	0,8	0,82
Accuracy	**0,7857**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 400 yaitu sebesar 78,57 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

5. Maksimal Iterasi 500

Hasil dari skenario pengujian 2 dengan maksimal iterasi 500 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.41 berikut.

Tabel 4.41 Hasil Pengujian Maksimal Iterasi 500

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	2	-1
8	2	2	0
9	2	2	0
10	2	2	0
11	2	2	0
12	2	2	0
13	2	2	0
14	2	2	0
15	2	2	0
16	2	3	-1
17	2	2	0
18	3	3	0
19	3	3	0
20	3	3	0
21	3	3	0
22	3	3	0
23	3	3	0
24	3	1	2
25	3	3	0

26	3	3	0
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.42 berikut.

Tabel 4.42 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 500

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	6	0	1	7
	2	1	9	0	10
	3	0	1	10	11
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.43.

Tabel 4.43 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 500

	Kategori		
	1	**2**	**3**
Precision	0,86	0,9	0,91
Recall	0,86	0,9	0,91
Accuracy	**0,8929**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 500 yaitu sebesar 89,29 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

6. Maksimal Iterasi 600

Hasil dari skenario pengujian 2 dengan maksimal iterasi 600 menggunakan 28 data kecamatan dan toleransi *error* sebesar 0.00001 dapat dilihat pada tabel 4.44 berikut.

Tabel 4.44 Hasil Pengujian Maksimal Iterasi 600

Data Ke-	Target Output	Output Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	2	2	0
9	2	2	0
10	2	2	0
11	2	2	0
12	2	3	-1
13	2	2	0
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	3	3	0
19	3	3	0
20	3	3	0
21	3	3	0
22	3	3	0
23	3	3	0
24	3	3	0
25	3	3	0

26	3	3	0
27	3	3	0
28	3	3	0

Dari tabel pengujian diatas, kemudian dapat dihitung nilai *precision* dan *recall* serta tingkat keakurasiannya menggunakan metode *Confusion Matrix Multi-Class* dengan membandingkan data prediksi dan data aktualnya. Untuk lebih jelasnya dapat dilihat pada tabel 4.45 berikut.

Tabel 4.45 *Confusion Matrix Multi-Class* Dari Maksimal Iterasi 600

		Aktual			Total Prediksi
		1	**2**	**3**	
Prediksi	**1**	7	0	0	7
	2	0	9	0	9
	3	0	1	11	12
Total Aktual		7	10	11	N = 28

Untuk menghitung nilai *precision* dan *recall* dari setiap kategorinya dapat digunakan persamaan 2.18 dan 2.19, sementara untuk mendapatkan tingkat keakurasiannya digunakan persamaan 2.17. Berikut hasilnya dapat dilihat pada tabel 4.46.

Tabel 4.46 *Precision, Recall* dan *Accuracy* Dari Maksimal Iterasi 600

	Kategori		
	1	**2**	**3**
Precision	1	1	0,92
Recall	1	0,9	1
Accuracy	**0,9643**		

Tingkat keakurasian dari hasil pengujian menggunakan maksimal iterasi 600 yaitu sebesar 96,43 %. Tampilan *screenshot* program dari hasil pengujian ini dapat dilihat pada lampiran B.

7. Kesimpulan Skenario Pengujian 2

Dari hasil skenario pengujian 2 yang ditampilkan pada tabel 4.29 sampai tabel 4.46, didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.47 berikut.

Tabel 4.47 Hasil Skenario Pengujian 2

Iterasi Pembelajaran	Akurasi Pengujian	Toleransi *Error* Pembelajaran	RMSE Pengujian
100	39,29 %	0.00001	0.819325
200	53,57 %	0.00001	0.707641
300	67,86 %	0.00001	0.628972
400	78,57 %	0.00001	0.384244
500	89,29 %	0.00001	0.290872
600	96,43 %	0.00001	0.210081

Dari data hasil skenario pengujian diatas dapat disimpulkan bahwa semakin banyak nilai maksimal iterasi yang diterapkan maka tingkat keakurasian yang didapat menjadi semakin tinggi.

4.4 Kesimpulan Pengujian Metode

Berdasarkan hasil pengujian yang dilakukan pada skenario 1, dapat disimpulkan bahwa nilai toleransi *error* akan mempengaruhi banyaknya jumlah iterasi yang dilakukan dan juga tingkat keakurasiannya. Semakin kecil nilai toleransi yang diterapkan, maka jumlah iterasi dan tingkat keakurasian yang dihasilkan akan semakin besar, begitu pun sebaliknya apabila nilai toleransi yang diterapkan semakin besar maka jumlah iterasi dan tingkat keakurasiannya akan semakin kecil. Hal ini menandakan bahwa nilai toleransi *error* berbanding terbalik dengan nilai jumlah iterasi dan tingkat keakurasian yang dihasilkannya.

Pada skenario pengujian 2, kondisi yang lebih menitikberatkan pada jumlah iterasi yang dilakukan menghasilkan tingkat keakurasian paling tingginya pada pengujian dengan penerapan jumlah iterasi yang paling banyak. Dari hal tersebut dapat disimpulkan bahwa jumlah iterasi yang diterapkan berbanding lurus dengan tingkat keakurasian yang akan dihasilkan.

Dari kedua skenario pengujian yang telah dilakukan, pengujian 1 menghasilkan tingkat keakurasian paling tinggi sebesar 92,86 % dengan rata-rata tingkat akurasi sebesar 65,48 %, sementara pada pengujian 2 tingkat keakurasian paling tinggi menghasilkan nilai sebesar 96,43 % dengan rata-rata tingkat keakurasian 70,83 %. Berdasarkan hasil tersebut, dapat dilihat bahwa faktor penerapan iterasi lebih berpengaruh untuk menghasilkan tingkat keakurasian yang lebih tinggi dibandingkan dengan penerapan nilai toleransi *error*. Secara keseluruhan, dapat disimpulkan bahwa metode *Adaptive Neuro Fuzzy Inference System* (ANFIS) ini memang dapat diterapkan dengan baik pada pengklasifikasian tingkat potensi luas lahan pertanian karena telah menghasilkan nilai akurasi yang cukup tinggi.