BAB 4
IMPLEMENTASI DAN PERANCANGAN

1.1. Implementasi

Tahap implementasi merupakan tahap dimana perangkat lunak yang akan dibangun dibuat yang juga merupakan tahap selanjutnya dari tahap perancangan. Tahap yang dilakukan untuk mengimplementasikan perancangan berdasarkan hasil analisis yang dimengerti oleh komputer.

1.1.1. Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan sistem ini dapat dilihat pada tabel 4.1.

Tabel 4. 1 Spesifikasi Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Processor Intel Core i5 2,5 GHz
2	RAM	RAM 4 GB
3	VGA	VGA 2 GB
4	Monitor	Monitor 14”
5	Hard Disk yang tersedia	50 GB
6	Peranti pendukung	*Keyboard* dan *mouse*

1.1.2. Implementasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan sistem ini dapat dilihat pada tabel 4.2.

Tabel 4. 2 Spesifikasi Implementasi Perangkat Lunak

No	Spesifikasi Perangkat Lunak
1	*System Operasi* Windows 8.1
2	MySQL
3	XAMPP
4	Microsoft Visual Studio 2013
5	ArcGIS 10.2

1.1.3. Implementasi *Database*

Pembuatan *database* dilakukan dengan menggunakan DBMS *MySQL*. Implementasi *database* dalam bahasa SQL dapat dilihat pada tabel 4.3.

Tabel 4. 3 Implementasi *Database*

No	Nama Tabel	SQL
1	t_hasil	CREATE TABLE `t_hasil` (`id` int(3) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `target_output` int(2) NOT NULL, `output_jaringan` int(2) NOT NULL, `eror` int(2) NOT NULL, PRIMARY KEY (`id`), KEY `id_parameter` (`id_parameter`), CONSTRAINT `t_hasil_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
2	t_lapisan1	CREATE TABLE `t_lapisan1` (`id_lapisan1` int(3) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `ch_rendah` decimal(30,8) NOT NULL, `ch_sedang` decimal(30,8) NOT NULL, `ch_tinggi` decimal(30,8) NOT NULL, `kl_rendah` decimal(30,8) NOT NULL, `kl_sedang` decimal(30,8) NOT NULL, `kl_tinggi` decimal(30,8) NOT NULL, `kt_rendah` decimal(30,8) NOT NULL, `kt_sedang` decimal(30,8) NOT NULL, `kt_tinggi` decimal(30,8) NOT NULL, PRIMARY KEY (`id_lapisan1`), KEY `kelurahan` (`id_parameter`), CONSTRAINT `t_lapisan1_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
3	t_lapisan2	CREATE TABLE `t_lapisan2` (`id_lapisan2` int(3) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `W1` decimal(30,8) NOT NULL, `W2` decimal(30,8) NOT NULL, `W3` decimal(30,8) NOT NULL, PRIMARY KEY (`id_lapisan2`), KEY `kelurahan` (`id_parameter`), CONSTRAINT `t_lapisan2_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;

No	Nama Tabel	SQL
4	t_lapisan3	CREATE TABLE `t_lapisan3` (`id_lapisan3` int(3) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `W1` decimal(30,8) NOT NULL, `W2` decimal(30,8) NOT NULL, `W3` decimal(30,8) NOT NULL, PRIMARY KEY (`id_lapisan3`), KEY `id_t2` (`id_parameter`), CONSTRAINT `t_lapisan3_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
5	t_lapisan4	CREATE TABLE `t_lapisan4` (`id_lapisan4` int(2) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `W1f1` decimal(30,8) DEFAULT NULL, `W2f2` decimal(30,8) DEFAULT NULL, `W3f3` decimal(30,8) DEFAULT NULL, PRIMARY KEY (`id_lapisan4`), KEY `kelurahan` (`id_parameter`), CONSTRAINT `t_lapisan4_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
6	t_lapisan5	CREATE TABLE `t_lapisan5` (`id_lapisan5` int(3) NOT NULL AUTO_INCREMENT, `id_parameter` int(3) NOT NULL, `sigma` decimal(30,8) NOT NULL, PRIMARY KEY (`id_lapisan5`), KEY `id_t4` (`id_parameter`), CONSTRAINT `t_lapisan5_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=latin1;
7	t_parameter	CREATE TABLE `t_parameter` (`id_parameter` int(3) NOT NULL AUTO_INCREMENT, `kelurahan` varchar(25) NOT NULL, `curah_hujan` decimal(6,1) NOT NULL, `kemiringan_lereng` decimal(6,1) NOT NULL, `ketinggian_tanah` decimal(6,1) NOT NULL, `target_output` decimal(6,1) NOT NULL, PRIMARY KEY (`id_parameter`)) ENGINE=InnoDB AUTO_INCREMENT=69 DEFAULT CHARSET=latin1;

No	Nama Tabel	SQL
8	t_parameter_premis	CREATE TABLE `t_parameter_premis` (`id_parameter_premis` varchar(20) NOT NULL, `a_rendah` decimal(30,8) NOT NULL, `b_rendah` decimal(30,8) NOT NULL, `c_rendah` decimal(30,8) NOT NULL, `a_sedang` decimal(30,8) NOT NULL, `b_sedang` decimal(30,8) NOT NULL, `c_sedang` decimal(30,8) NOT NULL, `a_tinggi` decimal(30,8) NOT NULL, `b_tinggi` decimal(30,8) NOT NULL, `c_tinggi` decimal(30,8) NOT NULL, PRIMARY KEY (`id_parameter_premis`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9	t_rmse	CREATE TABLE `t_rmse` (`id_rmse` int(3) NOT NULL AUTO_INCREMENT, `rmse` decimal(30,8) NOT NULL, PRIMARY KEY (`id_rmse`), KEY `id_lse` (`rmse`)) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;
10	t_detail_hasil	CREATE TABLE `t_detail_hasil` (`id` int(3) NOT NULL, `id_rmse` int(3) NOT NULL, KEY `id` (`id`), KEY `id_rmse` (`id_rmse`), CONSTRAINT `t_detail_hasil_ibfk_1` FOREIGN KEY (`id`) REFERENCES `t_hasil` (`id`), CONSTRAINT `t_detail_hasil_ibfk_2` FOREIGN KEY (`id_rmse`) REFERENCES `t_rmse` (`id_rmse`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
11	t_detail_parameter	CREATE TABLE `t_detail_parameter` (`id_parameter` int(3) NOT NULL, `id_parameter_premis` varchar(20) NOT NULL, KEY `id_parameter` (`id_parameter`), KEY `id_parameter_premis` (`id_parameter_premis`), CONSTRAINT `t_detail_parameter_ibfk_1` FOREIGN KEY (`id_parameter`) REFERENCES `t_parameter` (`id_parameter`), CONSTRAINT `t_detail_parameter_ibfk_2` FOREIGN KEY (`id_parameter_premis`) REFERENCES `t_parameter_premis` (`id_parameter_premis`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

1.1.4. Implementasi Antarmuka

Berikut merupakan implementasi antarmuka pada sistem ini, yang dapat dilihat pada tabel 4.4.

Tabel 4. 4 Implementasi Antarmuka

Nama	Deskripsi	Nama File
Home	Digunakan untuk menampilkan menu utama atau halaman awal dari program.	Form1.cs
Pengolahan anfis	Digunakan untuk menampilkan hasil perhitungan dalam pembelajaran ANFIS.	OlahANFIS.cs
Olah data parameter	Digunakan untuk menampilkan menu pengolahan data parameter yaitu import data, ubah data dan hapus data.	OlahDataParameter.cs
Olah data parameter premis	Digunakan untuk menampilkan menu pengolahan data parameter premis yaitu ubah data parameter premis.	OlahDataParameterPremis.cs
Ubah parameter	Digunakan untuk menampilkan data parameter yang dipilih untuk diubah.	UbahParameter.cs
Ubah parameter premis	Digunakan untuk menampilkan data parameter premis yang dipilih untuk diubah	UbahParameterPremis.cs
Uji coba anfis	Digunakan untuk menampilkan hasil perhitungan pengolahan anfis dan melihat hasil ujicoba prediksi banjir berdasarkan proses pembelajaran	UjiCoba.cs

Untuk tampilan gambar antarmuka dapat dilihat pada Lampiran A.

1.2. Pengujian Metode

Pengujian metode bertujuan untuk mengetahui performansi dari metode *Adaptive Neuro Fuzzy Inference System* (ANFIS) yang diimplementasikan untuk menentukan tingkat potensi longsor menggunakan data tematik. Pengujian metode ini menggunakan data pembelajaran 60% atau sebanyak 41 data kelurahan dan data uji 40% atau sebanyak 27 data kelurahan. Data uji yang digunakan terbagi dari 9 kelurahan rawan longsor tinggi, 9 kelurahan rawan longsor sedang dan 9 kelurahan tidak rawan longsor.

Pada metode ANFIS terdapat 2 buah *variable* yang dapat diubah, yaitu banyak iterasi dan toleransi *error*. Perubahan banyak iterasi dan toleransi *error* dilakukan untuk membatasi proses pembelajaran metode ANFIS.

Akurasi dari metode ANFIS ini didapat dari banyaknya hasil prediksi sistem yang sesuai dengan target output dan hasil prediksi sistem yang tidak sesuai. Jumlah hasil prediksi yang akurat dan jumlah hasil prediksi yang tidak akurat diubah menjadi nilai persen untuk memudahkan dalam mengetahui nilai akurasi dari metode ANFIS.

Pada kasus ini perhitungan akurasi menggunakan metode *Confusion Matrix* yang dimana menggunakan variable a,b,c,d untuk menghitung nilai akurasi data hasil pengolahan metode ANFIS yang menggunakan persamaan 2.15. *Confusion Matrix* digunakan dalam pengujian penerapan algoritma ANFIS dalam menentukan tingkat potensi longsor yang ditujukan untuk menguji hasil *output* yang dikeluarkan dengan target *output* yang ada, agar dapat menentukan mana hasil *output* yang tepat sesuai dengan kenyataan yang ada dan mana yang tidak sesuai kenyataan.

1.2.1. Pengujian 1

Pada pengujian 1 dilakukan menggunakan 27 data kelurahan uji coba, dimana 9 kelurahan rawan longsor tinggi, 9 kelurahan rawan longsor sedang dan 9 kelurahan tidak rawan longsor dengan menggunakan toleransi *error* dari nilai 0.0014, 0.0013, 0.0012, 0.0011, 0.0010 dengan banyak iterasi sebanyak 500.

1. Toleransi *error* 0.0014

Pengujian 1 dengan toleransi *error* 0.0014 yang menggunakan 27 data kelurahan dan iterasi sebanyak 500 dapat dilihat pada tabel 4.5, dan *matrix confusion* dari hasil toleransi *error* 0.0014 dapat dilihat pada tabel 4.6.

Tabel 4. 5 Hasil Toleransi *Error* 0.0014

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	2	-1
2	1	0	1
3	1	-1	2
4	1	1	0
5	1	0	1
6	1	0	1
7	1	2	-1
8	1	1	0
9	1	2	-1
10	2	1	1
11	2	1	1

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
12	2	2	0
13	2	1	1
14	2	1	1
15	2	1	1
16	2	3	-1
17	2	3	-1
18	2	1	1
19	3	4	-1
20	3	4	-1
21	3	2	1
22	3	2	1
23	3	2	1
24	3	3	0
25	3	4	-1
26	3	3	0
27	3	3	0

Tabel 4. 6 *Matrix Confusion* dari Hasil Toleransi *Error* 0.0014

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	21
	Positive	0	6

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0014 menghasilkan nilai akurasi sebesar 22,22%.

2. Toleransi *error* 0.0013

Pengujian 1 dengan toleransi *error* 0.0013 yang menggunakan 27 data kelurahan dan iterasi sebanyak 500 dapat dilihat pada tabel 4.7, dan *matrix confusion* dari hasil toleransi *error* 0.0013 dapat dilihat pada tabel 4.8.

Tabel 4. 7 Hasil Toleransi *Error* 0.0013

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	0	1
2	1	1	0
3	1	-1	2
4	1	1	0
5	1	0	1

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
6	1	0	1
7	1	1	0
8	1	1	0
9	1	2	-1
10	2	1	1
11	2	1	1
12	2	2	0
13	2	1	1
14	2	1	1
15	2	2	0
16	2	3	-1
17	2	3	-1
18	2	1	1
19	3	3	0
20	3	4	-1
21	3	2	1
22	3	2	1
23	3	3	0
24	3	3	0
25	3	4	-1
26	3	3	0
27	3	3	0

Tabel 4. 8 *Matrix Confusion* dari Hasil Toleransi *Error* 0.0013

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	16
	Positive	0	11

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0013 menghasilkan nilai akurasi sebesar 40,74%.

3. Toleransi *error* 0.0012

Pengujian 1 dengan toleransi *error* 0.0012 yang menggunakan 27 data kelurahan dan iterasi sebanyak 500 dapat dilihat pada tabel 4.9, dan *matrix confusion* dari hasil toleransi *error* 0.0012 dapat dilihat pada tabel 4.10.

Tabel 4. 9 Hasil Toleransi *Error* 0.0012

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	0	1
4	1	1	0
5	1	1	0
6	1	1	0
7	1	2	-1
8	1	1	0
9	1	2	-1
10	2	1	1
11	2	1	1
12	2	2	0
13	2	2	0
14	2	1	1
15	2	2	0
16	2	3	-1
17	2	2	0
18	2	2	0
19	3	3	0
20	3	4	-1
21	3	2	1
22	3	3	0
23	3	3	0
24	3	3	0
25	3	2	1
26	3	2	1
27	3	3	0

Tabel 4. 10 *Matrix Confusion* dari Hasil Toleransi *Error* 0.0012

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	11
	Positive	0	16

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0012 menghasilkan nilai akurasi sebesar 59,26%.

4. Toleransi *error* 0.0011

Pengujian 1 dengan toleransi *error* 0.0011 yang menggunakan 27 data kelurahan dan iterasi sebanyak 500 dapat dilihat pada tabel 4.11, dan *matrix confusion* dari hasil toleransi *error* 0.0011 dapat dilihat pada tabel 4.12.

Tabel 4. 11 Hasil Toleransi *error* 0.0011

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	0	1
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	2	-1
10	2	1	1
11	2	2	0
12	2	2	0
13	2	1	1
14	2	2	0
15	2	2	0
16	2	3	-1
17	2	2	0
18	2	1	1
19	3	3	0
20	3	4	-1
21	3	2	1
22	3	3	0
23	3	3	0
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 12 *Matrix Confusion* dari Hasil Toleransi *Error* 0.0011

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	8
	Positive	0	19

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0011 menghasilkan nilai akurasi sebesar 70,37%.

5. Toleransi *error* 0.0010

Pengujian 1 dengan toleransi *error* 0.0010 yang menggunakan 27 data kelurahan dan iterasi sebanyak 500 dapat dilihat pada tabel 4.13, dan *matrix confusion* dari hasil toleransi *error* 0.0010 dapat dilihat pada tabel 4.14.

Tabel 4. 13 Hasil Toleransi *error* 0.0010

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

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 14 *Matrix Confusion* dari Hasil Toleransi *Error* 0.0010

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	4
	Positive	0	23

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0010 menghasilkan nilai akurasi sebesar 85,19%.

Dari hasil pengujian pada tabel 4.14, maka didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.15.

Tabel 4. 15 Hasil Pengujian 1

Toleransi *error* pembelajaran	Akurasi pengujian	Iterasi terakhir pembelajaran	RMSE pengujian
0.0014	22.22%	59	0.925922
0.0013	40.74%	78	0.762947
0.0012	44.44%	102	0.728921
0.0011	70.37%	131	0.516844
0.0010	85.19%	185	0.365289

Dari data hasil pengujian 1 dapat dilihat jika semakin kecil toleransi *error* yang diberikan maka iterasi yang didapat akan semakin besar.

1.2.2. Pengujian 2

Pada pengujian 2 dilakukan menggunakan 27 data kelurahan uji coba, dimana 9 kelurahan rawan longsor tinggi, 9 kelurahan rawan longsor sedang dan 9 kelurahan tidak rawan longsor dengan menggunakan maksimal iterasi dari nilai 100, 200, 300, 400 dan 500 dengan nilai toleransi *error* 0.0001.

1. Banyak iterasi 100

Pengujian 2 dengan maksimal iterasi 100 yang menggunakan 27 data kelurahan dan toleransi *error* 0.0001 dapat dilihat pada tabel 4.16, dan *matrix confusion* dari hasil maksimal iterasi 100 dapat dilihat pada tabel 4.17.

Tabel 4. 16 Hasil Maksimal Iterasi 100

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

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
26	3	2	1
27	3	4	-1

Tabel 4. 17 *Matrix Confusion* dari Hasil Maksimal Iterasi 100

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	20
	Positive	0	7

Pada hasil pengujian dengan menggunakan maksimal iterasi 100 menghasilkan nilai akurasi sebesar 25,93%.

2. Banyak iterasi 200

Pengujian 2 dengan maksimal iterasi 200 yang menggunakan 27 data kelurahan dan toleransi *error* 0.0001 dapat dilihat pada tabel 4.18, dan *matrix confusion* dari hasil maksimal iterasi 200 dapat dilihat pada tabel 4.19.

Tabel 4. 18 Hasil Maksimal Iterasi 200

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

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
19	3	2	1
20	3	3	0
21	3	2	1
22	3	2	1
23	3	3	0
24	3	2	1
25	3	3	0
26	3	3	0
27	3	4	-1

Tabel 4. 19 *Matrix Confusion* dari Hasil Maksimal Iterasi 200

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	16
	Positive	0	11

Pada hasil pengujian dengan menggunakan maksimal iterasi 200 menghasilkan nilai akurasi sebesar 40,74%.

3. Banyak iterasi 300

Pengujian 2 dengan maksimal iterasi 300 yang menggunakan 27 data kelurahan dan toleransi *error* 0.0001 dapat dilihat pada tabel 4.20, dan *matrix confusion* dari hasil maksimal iterasi 300 dapat dilihat pada tabel 4.21.

Tabel 4. 20 Hasil Maksimum Iterasi 300

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	0	1
5	1	1	0
6	1	0	1
7	1	1	0
8	1	1	0
9	1	1	0
10	2	1	1

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
11	2	2	0
12	2	2	0
13	2	1	1
14	2	1	1
15	2	2	0
16	2	1	1
17	2	2	0
18	2	2	0
19	3	3	0
20	3	3	0
21	3	2	1
22	3	3	0
23	3	3	0
24	3	3	0
25	3	3	0
26	3	2	1
27	3	2	1

Tabel 4. 21 *Matrix Confusion* dari Hasil Maksimal Iterasi 300

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	9
	Positive	0	18

Pada hasil pengujian dengan menggunakan maksimal iterasi 300 menghasilkan nilai akurasi sebesar 66,67%.

4. Banyak iterasi 400

Pengujian 2 dengan maksimal iterasi 400 yang menggunakan 27 data kelurahan dan toleransi *error* 0.0001 dapat dilihat pada tabel 4.22, dan *matrix confusion* dari hasil maksimal iterasi 400 dapat dilihat pada tabel 4.23.

Tabel 4. 22 Hasil Maksimum Iterasi 400

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
3	1	1	0
4	1	0	1
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0
10	2	1	1
11	2	2	0
12	2	2	0
13	2	1	1
14	2	2	0
15	2	2	0
16	2	1	1
17	2	2	0
18	2	2	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	2	1
27	3	2	1

Tabel 4. 23 *Matrix Confusion* dari Hasil Maksimal Iterasi 400

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	6
	Positive	0	21

Pada hasil pengujian dengan menggunakan maksimal iterasi 400 menghasilkan nilai akurasi sebesar 77,78%.

5. Banyak iterasi 500

Pengujian 2 dengan maksimal iterasi 500 yang menggunakan 27 data kelurahan dan toleransi *error* 0.0001 dapat dilihat pada tabel 4.24, dan *matrix confusion* dari hasil maksimal iterasi 500 dapat dilihat pada tabel 4.25.

Tabel 4. 24 Hasil Maksimum 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	1	0
8	1	1	0
9	1	1	0
10	2	2	0
11	2	2	0
12	2	2	0
13	2	3	-1
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	2	2	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	2	1
27	3	3	0

Tabel 4. 25 *Matrix Confusion* dari Hasil Maksimal Iterasi 500

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	2
	Positive	0	25

Pada hasil pengujian dengan menggunakan maksimal iterasi 500 menghasilkan nilai akurasi sebesar 92,59%.

Dari hasil pengujian pada tabel 4.25, maka didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.26.

Tabel 4. 26 Hasil Pengujian 2

Iterasi pembelajaran	**Akurasi pengujian**	**Toleransi *error* pembelajaran**	**RMSE pengujian**
100	25.93%	0.0001	0.839203
200	40.74%	0.0001	0.681283
300	66.67%	0.0001	0.421284
400	77.78%	0.0001	0.353722
500	92.59%	0.0001	0.241207

Dari data hasil pengujian 2 dapat dilihat jika semakin banyak iterasi yang diberikan maka akurasi yang didapat akan semakin besar.

1.2.3. Pengujian 3

Pada pengujian 3 dilakukan menggunakan 27 data kelurahan uji coba, dimana 9 kelurahan rawan longsor tinggi, 9 kelurahan rawan longsor sedang dan 9 kelurahan tidak rawan longsor dengan hanya menggunakan maksimal iterasi dari nilai 420, 440, 460, 480, 500.

1. Banyak Iterasi 420

Pengujian 3 dengan maksimal iterasi 420 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.27, dan *matrix confusion* dari hasil maksimum iterasi 420 dapat dilihat pada tabel 4.28.

Tabel 4. 27 Hasil Maksimum Iterasi 420

Data Ke-	**Target *Output***	***Output* Jaringan**	***Error***
1	1	1	0
2	1	1	0
3	1	1	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
4	1	0	1
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	0	1
10	2	1	1
11	2	2	0
12	2	2	0
13	2	2	0
14	2	2	0
15	2	2	0
16	2	1	1
17	2	2	0
18	2	2	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	2	1
27	3	2	1

Tabel 4. 28 *Matrix Confusion* dari Hasil Maksimal Iterasi 420

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	6
	Positive	0	21

Pada hasil pengujian dengan menggunakan maksimal iterasi 420 menghasilkan nilai akurasi sebesar 77,78%.

2. Banyak Iterasi 440

Pengujian 3 dengan maksimal iterasi 440 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.29, dan *matrix confusion* dari hasil maksimum iterasi 440 dapat dilihat pada tabel 4.30.

Tabel 4. 29 Hasil Maksimum Iterasi 440

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0
4	1	0	1
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0
10	2	1	1
11	2	2	0
12	2	2	0
13	2	2	0
14	2	2	0
15	2	2	0
16	2	1	1
17	2	2	0
18	2	2	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	2	1
27	3	2	1

Tabel 4. 30 *Matrix Confusion* dari Hasil Maksimal Iterasi 440

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	5
	Positive	0	22

Pada hasil pengujian dengan menggunakan maksimal iterasi 440 menghasilkan nilai akurasi sebesar 81,48%.

3. Banyak Iterasi 460

Pengujian 3 dengan maksimal iterasi 460 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.31, dan *matrix confusion* dari hasil maksimum iterasi 460 dapat dilihat pada tabel 4.32.

Tabel 4. 31 Hasil Maksimum Iterasi 460

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

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
24	3	3	0
25	3	3	0
26	3	2	1
27	3	3	0

Tabel 4. 32 *Matrix Confusion* dari Hasil Maksimal Iterasi 460

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	4
	Positive	0	23

Pada hasil pengujian dengan menggunakan maksimal iterasi 460 menghasilkan nilai akurasi sebesar 85,19%.

4. Banyak iterasi 480

Pengujian 3 dengan maksimal iterasi 480 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.33, dan *matrix confusion* dari hasil maksimum iterasi 480 dapat dilihat pada tabel 4.34.

Tabel 4. 33 Hasil Maksimum Iterasi 480

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

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
18	2	2	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	2	1
27	3	3	0

Tabel 4. 34 *Matrix Confusion* dari Hasil Maksmial Iterasi 480

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	4
	Positive	0	23

Pada hasil pengujian dengan menggunakan maksimal iterasi 480 menghasilkan nilai akurasi sebesar 85,19%.

5. Banyak iterasi 500

Pengujian 3 dengan maksimal iterasi 500 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.35, dan *matrix confusion* dari hasil maksimum iterasi 500 dapat dilihat pada tabel 4.36.

Tabel 4. 35 Hasil 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	1	0
8	1	1	0
9	1	1	0
10	2	2	0
11	2	2	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
12	2	2	0
13	2	3	-1
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	2	2	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	2	1
27	3	3	0

Tabel 4. 36 *Matrix Confusion* dari Hasil Maksimal Iterasi 500

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	2
	Positive	0	25

Pada hasil pengujian dengan menggunakan maksimal iterasi 500 menghasilkan nilai akurasi sebesar 92,59%.

Dari hasil pengujian 3 pada tabel 4.36, maka didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.37.

Tabel 4. 37 Hasil Pengujian 3

Iterasi pembelajaran	Akurasi pengujian	RMSE pengujian
420	77.78%	0.353433
440	81.48%	0.310487
460	85.19%	0.272894
480	85.19%	0.281929
500	92.59%	0.241207

Rata-rata akurasi dari pengujian ke-3 adalah 84,45% dengan akurasi tertinggi adalah 92,59%.

1.2.4. Pengujian 4

Pada pengujian 4 dilakukan menggunakan 27 data kelurahan uji coba, dimana 9 kelurahan rawan longsor tinggi, 9 kelurahan rawan longsor sedang dan 9 kelurahan tidak rawan longsor dengan hanya menggunakan toleransi *error* dari nilai 0.0005, 0.0004, 0.0003, 0.0002, 0.0001.

1. Toleransi *error* 0.0005

Pengujian 4 dengan toleransi *error* 0.0005 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.38, dan *matrix confusion* dari hasil toleransi *error* 0.0005 dapat dilihat pada tabel 4.39.

Tabel 4. 38 Hasil Toleransi *error* 0.0005

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	1	1	0
9	1	0	1
10	2	1	1
11	2	2	0
12	2	2	0
13	2	1	1
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	2	1	1
19	3	3	0
20	3	3	0
21	3	2	1

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
22	3	3	0
23	3	3	0
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 39 *Matrix Confusion* dari Hasil Toleransi *error* 0.0005

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	5
	Positive	0	22

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0005 menghasilkan nilai akurasi sebesar 81,48%.

2. Toleransi *error* 0.0004

Pengujian 4 dengan toleransi *error* 0.0004 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.40, dan *matrix confusion* dari hasil toleransi *error* 0.0004 dapat dilihat pada tabel 4.41.

Tabel 4. 40 Hasil Toleransi *error* 0.0004

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	0	1
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0
10	2	1	1
11	2	2	0
12	2	2	0
13	2	2	0
14	2	2	0
15	2	2	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
16	2	2	0
17	2	2	0
18	2	1	1
19	3	3	0
20	3	3	0
21	3	2	1
22	3	3	0
23	3	3	0
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 41 *Matrix Confusion* dari Hasil Toleransi *error* 0.0004

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	4
	Positive	0	23

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0004 menghasilkan nilai akurasi sebesar 85,19%.

3. Toleransi *error* 0.0003

Pengujian 4 dengan toleransi *error* 0.0003 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.42, dan *matrix confusion* dari hasil toleransi *error* 0.0003 dapat dilihat pada tabel 4.43.

Tabel 4. 42 Hasil Toleransi *error* 0.0003

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	0	1
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	1	1	0
9	1	1	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
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	2	2	0
19	3	3	0
20	3	3	0
21	3	2	1
22	3	3	0
23	3	2	1
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 43 *Matrix Confusion* dari Hasil Toleransi *error* 0.0003

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	3
	Positive	0	24

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0003 menghasilkan nilai akurasi sebesar 88,89%.

4. Toleransi *error* 0.0002

Pengujian 4 dengan toleransi *error* 0.0002 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.44, dan *matrix confusion* dari hasil toleransi *error* 0.0002 dapat dilihat pada tabel 4.45.

Tabel 4. 44 Hasil Toleransi *error* 0.0002

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
1	1	1	0
2	1	1	0
3	1	1	0

Data Ke-	Target *Output*	*Output* Jaringan	*Error*
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0
9	1	1	0
10	2	1	1
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	2	2	0
19	3	3	0
20	3	2	1
21	3	3	0
22	3	3	0
23	3	2	1
24	3	3	0
25	3	3	0
26	3	3	0
27	3	3	0

Tabel 4. 45 *Matrix Confusion* dari Hasil Toleransi *error* 0.0002

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	4
	Positive	0	23

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0002 menghasilkan nilai akurasi sebesar 85,19%.

5. Toleransi *error* 0.0001

Pengujian 4 dengan toleransi *error* 0.0001 yang menggunakan 27 data kelurahan dapat dilihat pada tabel 4.46, dan *matrix confusion* dari hasil toleransi *error* 0.0001 dapat dilihat pada tabel 4.47.

Tabel 4. 46 Hasil Toleransi *error* 0.0001

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	1	1	0
9	1	1	0
10	2	2	0
11	2	2	0
12	2	2	0
13	2	1	1
14	2	2	0
15	2	2	0
16	2	2	0
17	2	2	0
18	2	2	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	2	1
27	3	3	0

Tabel 4. 47 *Matrix Confusion* dari Hasil Toleransi *error* 0.0001

		Prediksi	
		Negative	*Positive*
Aktual	*Negative*	0	2
	Positive	0	25

Pada hasil pengujian dengan menggunakan toleransi *error* 0.0001 menghasilkan nilai akurasi sebesar 92,59%.

Dari hasil pengujian 4 pada tabel 4.47, maka didapatkan hasil secara keseluruhan yang dapat dilihat pada tabel 4.48.

Tabel 4. 48 Hasil Pengujian 4

Toleransi *error* pembelajaran	**Akurasi pengujian**	**RMSE pengujian**
0.0005	81.48%	0.307589
0.0004	85.19%	0.287419
0.0003	88.89%	0.269344
0.0002	85.19%	0.275983
0.0001	92.59 %	0.239783

Rata-rata akurasi dari pengujian ke-4 adalah 86,67% dengan akurasi tertinggi adalah 92,59%.

1.3. Kesimpulan Pengujian

Berdasarkan hasil pengujian ke-1 dapat disimpulkan bahwa semakin besarnya toleransi *error* yang diberikan maka akan mempengaruhi banyaknya iterasi yang diperlukan. Sedangkan berdasarkan hasil pengujian ke-2 dapat disimpulkan bahwa semakin banyaknya iterasi yang diberikan maka akan mempengaruhi akurasi yang didapatkan akan semakin besar.

Berdasarkan hasil pengujian ke-3 dan ke-4 dapat disimpulkan bahwa toleransi *error* lebih mempengaruhi hasil akurasi dibandingkan dengan banyaknya iterasi / *epoch*. Dengan rata-rata akurasi dari pengujian toleransi *error* sebesar 86,67% dan akurasi tertinggi adalah 92,59%

Berdasarkan hasil pengujian ke 1,2,3 dan 4 algoritma *Adaptive Neuro Fuzzy Inference System* (ANFIS) dapat digunakan dalam menentukan tingkat potensi longsor, dengan akurasi terbesar sebesar 92.59% dengan rata-rata akurasi sebesar 86,67%.