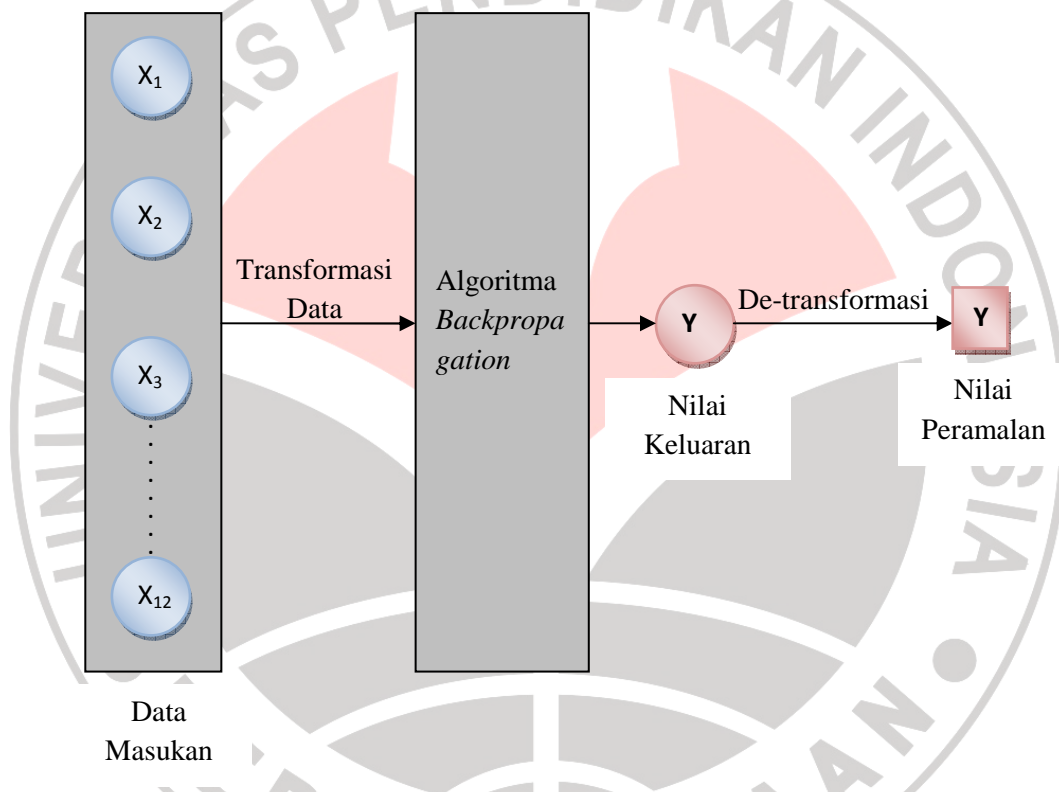


BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Tahapan Umum Peramalan Curah Hujan

Tahapan umum peramalan curah hujan dapat dilihat pada gambar 4.1 berikut.



Gambar 4.1 Tahapan Umum Peramalan Curah Hujan

Data masukan yang diperlukan yaitu data curah hujan pada bulan yang sama dalam satuan milimeter sebanyak 12 masukan. Sebagai contoh, ketika akan meramalkan data curah hujan bulan Desember 2011, maka data masukan yang diperlukan yaitu data curah hujan bulan Desember 1999, Desember 2000, Desember 2001, Desember 2002, Desember 2003, Desember 2004, Desember

2005, Desember 2006, Desember 2007, Desember 2008, Desember 2009, dan Desember 2010.

Data masukan kemudian ditransformasi ke dalam rentang 0-1. Kemudian dilakukan algoritma *backpropagation* untuk mendapatkan nilai keluaran (nilai peramalan). Nilai keluaran yang dihasilkan berupa nilai curah hujan untuk satu bulan ke depan. Berikutnya nilai keluaran di de-transformasi atau dilakukan transformasi kembali untuk menghasilkan nilai peramalan dalam satuan mm.

Agar sistem curah hujan yang dibuat dapat meramalkan nilai pada satu bulan ke depan, maka ada tahapan yang harus dilakukan. Tahapan-tahapan peramalan curah hujan dapat dilihat pada poin 4.2 dan 4.3 berikut.

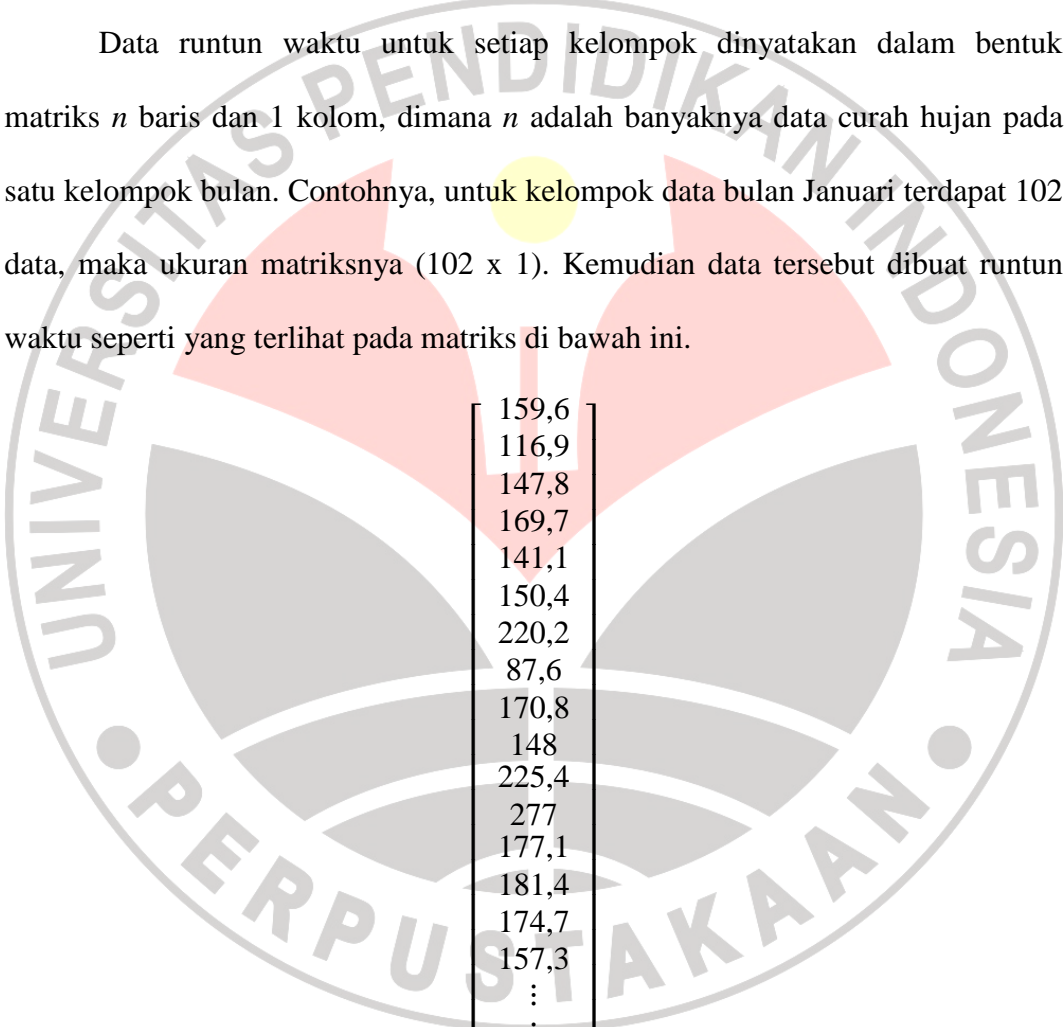
4.2 Representasi Masalah

Besarnya curah hujan dijabarkan sebagai banyaknya air yang mencapai permukaan tanah dalam waktu tertentu. Satuan waktu curah hujan yang dipakai pada penelitian ini adalah dalam bulan. Sedangkan untuk satuan jumlah curah hujannya dinyatakan dalam milimeter (mm).

Curah hujan satu milimeter artinya air hujan yang jatuh pada setiap permukaan seluas 1 m^2 setinggi 1 mm, dengan tidak menguap, meresap, atau mengalir. Dengan kata lain, sejumlah air hujan yang jatuh sebanyak 1 liter pada setiap luasan 1 m^2 (BMG, 2008). Sebagai contoh, data curah hujan bulan Januari 1910 tertulis 159,6 mm. Hal ini berarti, jumlah total curah hujan bulan Januari 1910 sebanyak 159,6 liter/ m^2 . Atau dapat juga diartikan, dalam 1 m^2 ketinggian curah hujannya adalah 159,6 mm.

Data curah hujan dinyatakan dalam *time series* (runtun waktu), $x_1, x_2, x_3, x_4, \dots, x_n$. Dari data curah hujan pada lampiran 1, maka data yang ada dikelompokkan berdasarkan bulan. Antara bulan satu dengan yang lainnya tidak ada keterkaitan runtun waktu. Data runtun waktu berlaku untuk masing-masing kelompok bulan.

Data runtun waktu untuk setiap kelompok dinyatakan dalam bentuk matriks n baris dan 1 kolom, dimana n adalah banyaknya data curah hujan pada satu kelompok bulan. Contohnya, untuk kelompok data bulan Januari terdapat 102 data, maka ukuran matriksnya (102 x 1). Kemudian data tersebut dibuat runtun waktu seperti yang terlihat pada matriks di bawah ini.



159,6
116,9
147,8
169,7
141,1
150,4
220,2
87,6
170,8
148
225,4
277
177,1
181,4
174,7
157,3
⋮
⋮
⋮
163,1

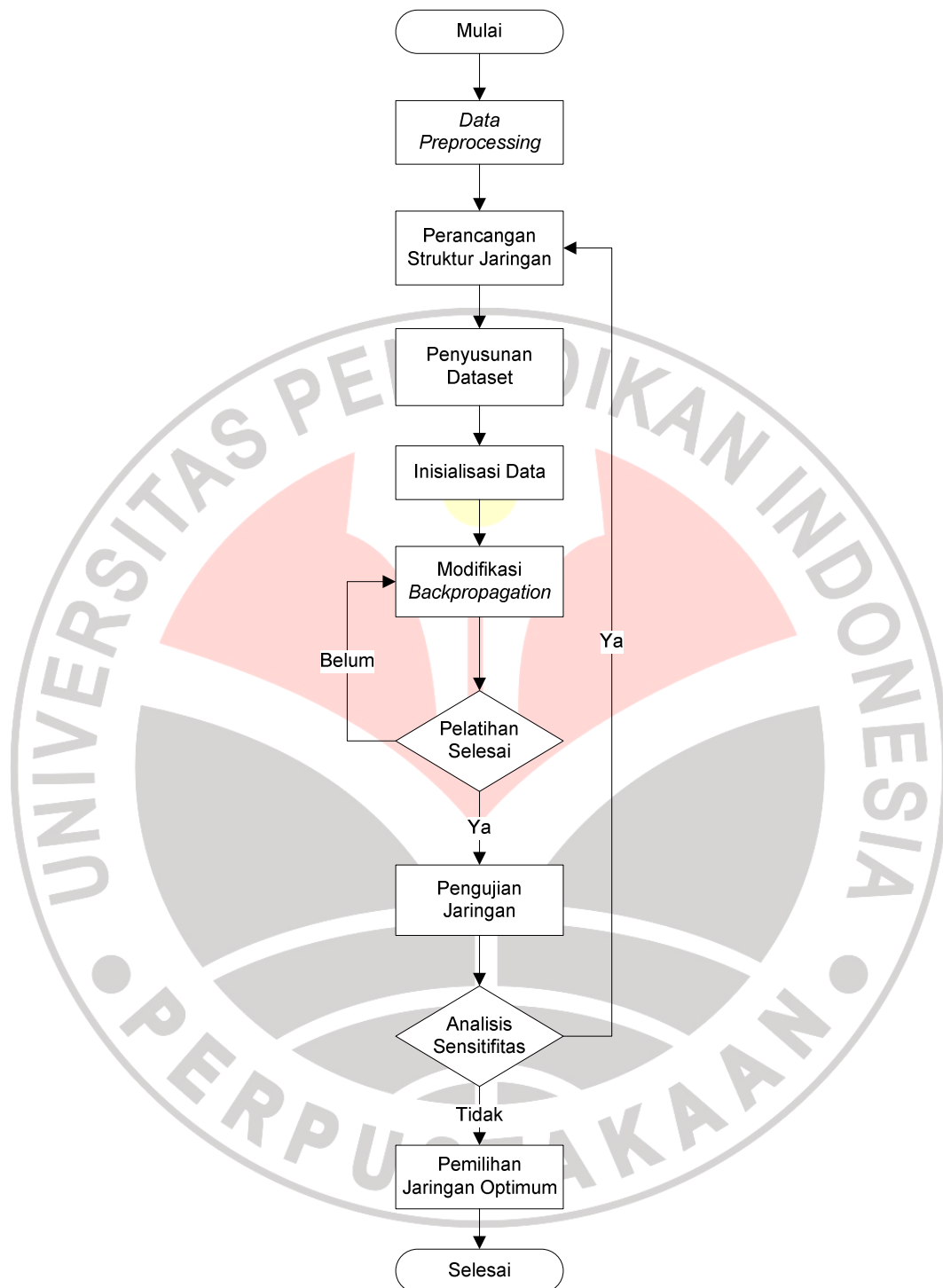
Data runtun waktu curah hujan untuk masing-masing bulan kemudian disimpan dalam bentuk file teks (.txt). Apabila terdapat bilangan desimal, maka koma diganti titik agar dapat diproses sistem.

4.3 Pengembangan Sistem Peramalan Curah Hujan

Dalam pengembangan sistem peramalan curah hujan menggunakan metode *backpropagation*, ada beberapa tahap yang perlu dilakukan, yaitu sebagai berikut.

- a. *Data preprocessing*.
- b. Perancangan struktur jaringan.
- c. Penyusunan data set pelatihan dan pengujian.
- d. Inisialisasi data.
- e. Modifikasi algoritma pelatihan *Backpropagation*.
- f. Pengujian jaringan.
- g. Analisis sensitifitas.
- h. Pemilihan jaringan optimum untuk peramalan curah hujan.

Tahapan-tahapan tersebut merupakan suatu proses yang dapat berulang. Proses tersebut dapat dilihat pada *flowchart* gambar 4.2 berikut. Sedangkan untuk penjelasan masing-masing tahapan dijelaskan pada poin 4.3.1 sampai 4.3.8.



Gambar 4.2 *Flowchart* Pengembangan Sistem

4.3.1 *Data Preprocessing*

Data preprocessing (pra proses data) dilakukan agar data yang akan digunakan dalam proses peramalan benar, tidak mengandung kesalahan, dan sesuai dengan algoritma yang digunakan. Adapun tahapan yang dilakukan yaitu sebagai berikut.

- a. *Data cleansing* (pembersihan data) yang di dalamnya terdapat penanganan data yang hilang dan pendeteksian *outlier*.
- b. Transformasi data.
- c. Pembagian data curah hujan.

Penjelasan lebih lanjut mengenai *data preprocessing* dapat dilihat pada poin 4.3.1.1 sampai 4.3.1.3 berikut ini.

4.3.1.1 *Data Cleansing (Pembersihan Data)*

Pada *data cleansing* terdapat dua tahap yang perlu dilakukan, yaitu:

- a. Penanganan Data yang Hilang

Dalam mendeteksi data yang hilang, dapat dilihat pada data curah hujan yang ada di lampiran 1. Bila ada data yang hilang, maka pada data akan terlihat kosong. Bila demikian, maka data yang hilang diganti dengan nilai rata-rata curah hujan bulanannya. Dari hasil pengamatan, dapat disimpulkan bahwa tidak ada data yang hilang. Sehingga untuk penanganan data yang hilang tidak dilakukan.

b. Pendeteksian *Outlier*

Yang termasuk data *outlier* atau data asing di sini yaitu data yang kemungkinan salah karena proses memasukkan data (seperti data yang bernilai minus) dan data-data anomali atau data yang jumlahnya tidak normal (dapat lebih maupun kurang dari jumlah normal). Untuk mendeteksi *outlier* atau data asing, dilakukan dengan metode statistika seperti yang telah dijelaskan pada poin 2.4.2. Adapun tahapan yang dilakukan adalah sebagai berikut.

- 1) Untuk setiap kelompok bulan, lakukan langkah 2 sampai 6.
- 2) Urutkan data dari nilai yang terkecil sampai terbesar.
- 3) Hitung nilai Q1, Q3, IQR.

$$Q_i = \text{nilai data ke-} \left(\frac{i(n+1)}{4} \right) \dots\dots\dots(4.1)$$

dimana: $i = 1, 3$; $n =$ banyak data

$$IQR = Q_3 - Q_1 \dots\dots\dots(4.2)$$

- 4) Apabila ada nilai data yang berada di luar pagar dalam, maka dinyatakan sebagai *outlier*.

Pagar dalam terletak pada interval:

$$[Q_1 - 1,5 \cdot IQR \text{ sampai } Q_3 + 1,5 \cdot IQR] \dots\dots\dots(4.3)$$

- 5) Hitung rata-rata data curah hujan yang tidak mengandung nilai *outlier*.

$$\text{Rata-rata} = (x_1 + x_2 + x_3 + \dots + x_n) / n;$$

dimana: x_n adalah nilai data ke- n yang tidak mengandung *outlier*.

n adalah banyaknya data yang tidak mengandung *outlier*.

- 6) Apabila terdapat nilai *outlier* maka diganti dengan nilai rata-rata bulanannya.

Hasil perhitungannya adalah sebagai berikut:

- 1) Pengurutan data untuk setiap kelompok dapat dilihat pada lampiran 2.
- 2) Nilai Q1 terletak pada:

$$\begin{aligned} & \left(\frac{i(n+1)}{4} \right) \\ &= \left(\frac{1(102+1)}{4} \right) \\ &= 25,75 \\ &\approx 26 \end{aligned}$$

Pada kelompok data bulan Januari, data ke-26 bernilai 132,6. Dengan demikian nilai Q1 untuk kelompok bulan Januari = 132,6. Sedangkan untuk bulan Februari nilai Q1 = 70. Nilai Q1 kelompok bulan lainnya dapat dilihat pada lampiran 2.

- 3) Nilai Q3 terletak pada:

$$\begin{aligned} & \left(\frac{i(n+1)}{4} \right) \\ &= \left(\frac{3(102+1)}{4} \right) \\ &= 77,25 \\ &\approx 77 \end{aligned}$$

Pada kelompok data bulan Januari, data ke-77 bernilai 228,4. Dengan demikian nilai Q3 untuk kelompok bulan Januari = 228,4. Sedangkan untuk bulan Februari nilai Q3 = 169,7. Untuk lebih jelasnya, pendeteksian *outlier* dapat dilihat pada lampiran 2. Nilai Q1 kelompok bulan lainnya dapat dilihat pada lampiran 2.

- 4) Nilai *outlier* terdapat pada kelompok data bulan Maret, April, Mei, Juli, dan Agustus. Selanjutnya nilai *outlier* tersebut diganti dengan nilai rata-rata curah hujan masing-masing kelompok.

4.3.1.2 Transformasi Data

Transformasi data dimaksudkan untuk mengubah nilai data menjadi rentang tertentu yang disesuaikan dengan rentang nilai fungsi aktivasi yang digunakan pada algoritma *backpropagation*. Karena jumlah curah hujan tidak mungkin bernilai minus atau kurang dari 0, maka data ditransformasi dalam rentang 0 sampai 1 dengan rumus sebagai berikut.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \dots\dots\dots(4.4)$$

Dimana:

x' = nilai hasil transformasi

x = nilai aktual

x_{\min} = nilai minimum data kelompok

x_{\max} = nilai maksimum data kelompok

Transformasi data curah hujan dilakukan untuk masing-masing kelompok bulan. Nilai minimum dan maksimum yang dipakai merupakan nilai minimum dan maksimum masing-masing kelompok. Pada bulan Januari, nilai minimum data adalah 21,1 dan nilai maksimumnya 355,7.

Untuk data curah hujan bernilai 21,1 (minimum), maka transformasi datanya:

$$x' = \frac{21,1 - 21,1}{355,7 - 21,1} = 0$$

Untuk data curah hujan bernilai 355,7 (maksimum), maka:

$$x' = \frac{355,7 - 21,1}{355,7 - 21,1} = 1$$

Untuk data curah hujan bulan Januari 1910, nilainya 159,6, maka:

$$x' = \frac{159,6 - 21,1}{355,7 - 21,1} = 0,414$$

Transformasi seluruh data curah hujan dapat dilihat pada kurva transformasi curah hujan di lampiran 2.

4.3.1.3 Pembagian Data

Langkah terakhir *data preprocessing* yaitu membagi data menjadi dua bagian, data pelatihan dan pengujian. Untuk data pelatihan, digunakan 80% dari banyaknya data pada masing-masing kelompok bulan. Sedangkan untuk data pengujian sebanyak 20%.

Pada kelompok bulan Januari sampai Oktober, data pelatihan sebanyak 82 data (tahun 1910 - 1991) dan data pengujian sebanyak 20 data (tahun 1992 - 2011). Pada kelompok bulan November dan Desember, data pelatihan sebanyak 81 (tahun 1910 - 1990) dan data pengujian sebanyak 20 data (tahun 1991 - 2010).

Dengan data pelatihan, data diuji untuk melihat kemampuan jaringan dalam mengenali pola data yang diberikan. Sedangkan data pengujian digunakan untuk melihat kemampuan jaringan dalam meramalkan nilai curah hujan.

4.3.2 Perancangan Struktur Jaringan

Setelah *data preprocessing*, selanjutnya dilakukan perancangan struktur jaringan. Pada perancangan struktur jaringan terdapat tahap perancangan arsitektur jaringan dan menentukan fungsi aktivasi yang dijabarkan pada poin 4.3.2.1 dan 4.3.2.2 berikut ini.

4.3.2.1 Perancangan Arsitektur Jaringan

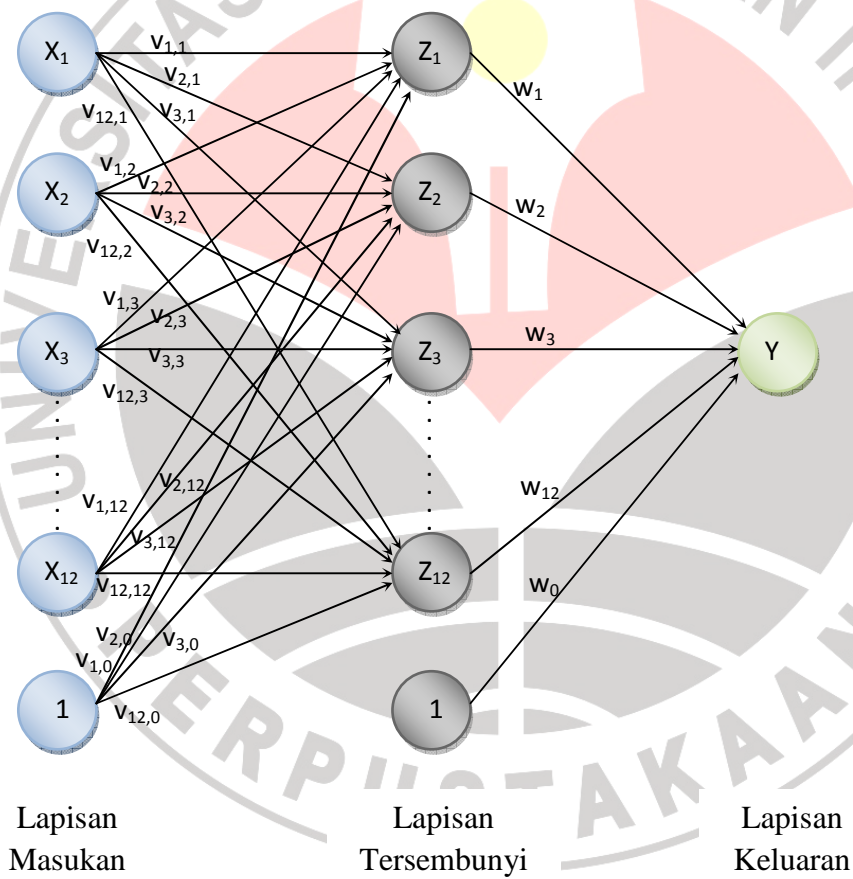
Dalam perancangan arsitektur jaringan yang dilakukan yaitu: penentuan jumlah simpul pada lapisan masukan, jumlah lapisan tersembunyi, jumlah simpul lapisan tersembunyi dan jumlah simpul keluaran yang akan digunakan dalam jaringan. Terdapat beberapa aturan yang dapat membantu perancangan jaringan yang optimum, yaitu sebagai berikut. (Siang, 2009)

- a. Jumlah simpul masukan sama dengan periode dimana data berfluktuasi.
- b. Jumlah simpul keluaran sama dengan bilangan keluaran masalah.
- c. Mulai dengan satu lapisan tersembunyi dan digunakan lebih dari satu lapisan tersembunyi jika diperlukan.
- d. Jika menggunakan satu lapisan tersembunyi, jumlah simpul tersembunyi adalah 75% dari jumlah simpul masukan. Atau dapat juga digunakan rentang 0,5 – 3 kali bilangan simpul masukan. (Palit, 2005)

Berdasarkan aturan tersebut, maka:

- Jumlah simpul masukan adalah 12.
- Jumlah simpul keluaran adalah satu karena data yang akan diramal adalah satu bulan berikutnya. Hal ini dimaksudkan agar hasil yang diperoleh lebih akurat.
- Lapisan tersembunyi adalah satu lapis dengan jumlah simpulnya 12. Tetapi pada saat studi kasus, jumlah simpul yang digunakan berbeda-beda.

Berikut ini merupakan gambar arsitektur jaringannya.



Gambar 4.3 Arsitektur Jaringan *Backpropagation*

4.3.2.2 Menentukan Fungsi Aktivasi

Fungsi aktivasi yang digunakan yaitu sigmoid biner yang dirumuskan sebagai berikut.

$$y = f(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(4.5)$$

$$\text{dengan fungsi turunannya } f'(x) = f(x) [1-f(x)] \dots\dots\dots(4.6)$$

Fungsi aktivasi sigmoid biner dipilih karena nilai keluaran yang dihasilkan terletak pada interval 0 sampai 1.

4.3.3 Menyusun Data Set Pelatihan dan Pengujian

Seperti yang telah dijelaskan sebelumnya, data curah hujan dinyatakan dalam *time series* (runtun waktu), $x_1, x_2, x_3, x_4, \dots, x_n$. Masalah peramalan yang dibahas yaitu menentukan nilai x_{n+1} berdasarkan x_1 sampai x_n . Hal ini berpengaruh terhadap data set pelatihan yang digunakan. Karena node masukan yang digunakan 12 dan node keluarannya satu, maka data set nya adalah sebagai berikut. (Malasri, 2000)

Tabel 4.1 Penyusunan Data Set

No.	Data Set	Target
1.	$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}$	x_{13}
2.	$x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}$	x_{14}
3.	$x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}$	x_{15}

dan seterusnya.

Apabila diterapkan pada data curah hujan yang telah dilakukan *preprocessing*, sebagai contoh, maka data set pelatihan untuk kelompok bulan Januari adalah sebagai berikut.

Tabel 4.2 Data Set Pelatihan

No.	Data Set	Target
1.	0,414 0,286 0,379 0,444 0,359 0,386 0,595 0,199 0,447 0,379 0,611 0,765	0,466
2.	0,286 0,379 0,444 0,359 0,386 0,595 0,199 0,447 0,379 0,611 0,765 0,466	0,479
3.	0,379 0,444 0,359 0,386 0,595 0,199 0,447 0,379 0,611 0,765 0,466 0,479	0,459
4.	0,444 0,359 0,386 0,595 0,199 0,447 0,379 0,611 0,765 0,466 0,479 0,459	0,407
5.	0,359 0,386 0,595 0,199 0,447 0,379 0,611 0,765 0,466 0,479 0,459 0,407	0,632
.	.	.
.	.	.
.	.	.
.	.	.
70.	0,345 0,265 0,518 0,480 0,695 0,686 0,143 0,538 0,169 0,632 0,556 0,817	0,454

Dengan demikian, terdapat 70 set data pelatihan. Sedangkan untuk data set pengujian terdapat 8 set data, yaitu:

Tabel 4.3 Data Set Pengujian

No.	Data Set	Target
1.	0,372 0,879 0,684 0,677 0,307 0,101 0,467 0,714 0,568 0,304 0,663 0,430	0,683
2.	0,879 0,684 0,677 0,307 0,101 0,467 0,714 0,568 0,304 0,663 0,430 0,683	0,694
3.	0,684 0,677 0,307 0,101 0,467 0,714 0,568 0,304 0,663 0,430 0,683	0,357
4.	0,677 0,307 0,101 0,467 0,714 0,568 0,304 0,663 0,430 0,683 0,357	0,733
5.	0,307 0,101 0,467 0,714 0,568 0,304 0,663 0,430 0,683 0,357 0,733	0,862
.	.	.
.	.	.
.	.	.
.	.	.

	.	.
8.	0,714 0,568 0,304 0,663 0,430 0,683 0,694 0,357 0,733 0,862 0,577 0,202	0,424

Pada data set pengujian, target digunakan untuk menghitung persentase kesalahan antara nilai keluaran jaringan (nilai peramalan) dengan nilai target yang diharapkan. Persentase kesalahan dihitung dengan MAPE yang akan dibahas secara detail pada tahap pengujian jaringan poin 4.3.6.

4.3.4 Inisialisasi Data

Yang termasuk ke dalam inisialisasi data yaitu: inisialisasi bobot, variabel *learning rate*, momentum, epoch maksimum, dan nilai batas toleransi yang dijelaskan berikut.

4.3.4.1 Inisialisasi Bobot

Untuk bobot lapisan tersembunyi, diinisialisasi dengan teknik Nguyen-Widrow. Penjelasan lebih lanjut mengenai teknik ini dapat dilihat pada Bab II poin 2.6.6.1. Berikut ini adalah pembahasan teknik inisialisasi bobotnya.

- a. Inisialisasi bobot lapisan tersembunyi (v_{ji}) dengan bilangan acak dari -0,5 sampai 0,5. Bobot lapisan tersembunyi disimbolkan dengan v_{ji} , dimana v merupakan nama bobotnya, j urutan simpul lapisan tersembunyi, dan i urutan simpul lapisan masukan. Sebagai contoh, berikut ini merupakan nilai inisialisasi bobot acaknya.

Tabel 4.4 Inisialisasi Bobot Bilangan Acak

		Simpul Lapisan Tersembunyi					
		1	2	3	4	5	6
Simpul Lapisan Masukan	1	0,315	0,457	0,179	0,195	0,209	0,251
	2	0,406	-0,015	0,258	-0,183	0,255	-0,245
	3	-0,373	0,300	0,243	0,450	-0,224	0,006
	4	0,413	-0,358	-0,108	-0,466	0,180	0,199
	5	0,132	-0,078	0,156	-0,061	0,155	0,391
	6	-0,403	0,416	-0,329	-0,118	-0,337	0,459
	7	-0,222	0,292	0,206	0,266	-0,381	0,047
	8	0,047	0,460	-0,468	0,295	-0,002	-0,361
	9	0,458	0,156	-0,223	-0,313	0,460	-0,351
	10	0,465	-0,464	-0,454	-0,010	-0,160	-0,243
	11	-0,342	0,349	-0,403	-0,054	0,085	0,341
	12	0,471	0,434	0,324	0,146	-0,276	-0,246
		7	8	9	10	11	12
Simpul Lapisan Masukan	1	0,314	0,417	0,069	0,189	-0,393	-0,318
	2	-0,257	-0,214	-0,031	0,248	0,462	-0,236
	3	0,429	0,257	-0,488	-0,050	-0,495	-0,355
	4	-0,150	0,254	-0,163	-0,416	0,275	-0,364
	5	-0,303	-0,120	-0,338	-0,271	0,317	0,369
	6	-0,249	0,068	0,294	0,413	0,369	0,080
	7	0,116	-0,424	-0,189	-0,348	-0,416	0,050
	8	-0,027	-0,446	0,029	0,326	-0,100	-0,355
	9	-0,148	0,031	-0,334	0,038	-0,240	0,353
	10	0,331	0,279	0,102	0,496	0,300	0,122
	11	0,085	0,434	-0,237	-0,422	-0,069	-0,149
	12	0,050	-0,370	0,154	-0,057	0,411	0,013

b. Untuk masing-masing simpul tersembunyi, hitung:

$$\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{j12}^2}$$

Untuk menghitung v_1 , maka:

$$\|v_1\| = \sqrt{(0,315)^2 + (0,406)^2 + (-0,373)^2 + (0,413)^2 + (0,132)^2 + (-0,403)^2 + (-0,222)^2 + (0,047)^2 +$$

$$(0,458)^2 + (0,465)^2 + (-0,342)^2 + (0,471)^2 \\ = 1,252$$

Setelah dilakukan perhitungan untuk simpul yang lainnya seperti di atas, maka diperoleh:

$$v_2=1,206; v_3=1,042; v_4=0,888; v_5=0,891; v_6=1,009; v_7=0,825; v_8=1,074; \\ v_9=0,843; v_{10}=1,083; v_{11}=1,196; v_{12}=0,921$$

- c. Selanjutnya menghitung faktor skala β , dengan rumus $(0,7 \sqrt[n]{p})$ dimana n adalah jumlah simpul masukan dan p adalah jumlah simpul tersembunyi. Maka nilai $\beta=(0,7 \sqrt[12]{12})=0,8611$. Kemudian dihitung bobot baru dengan rumus:

$$v_{ji} = \frac{\beta v_{ji}(\text{lama})}{||v_j||}$$

Sebagai contoh, bobot dari simpul lapisan masukan pertama ke simpul lapisan tersembunyi pertama (v_{11}), v_{12} , dan v_{21} dapat diperoleh dengan:

$$v_{11} = \frac{0,8611 * 0,315}{1,252} = 0,216$$

$$v_{12} = \frac{0,8611 * 0,406}{1,252} = 0,279$$

$$v_{21} = \frac{0,8611 * 0,475}{1,206} = 0,327$$

Untuk bias (v_{j0}), diinisialisasi dengan bilangan acak antara $-\beta$ dan β . Hasil perhitungan bobot baru beserta biasnya dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Inisialisasi Bobot Lapisan Tersembunyi

		Simpul Lapisan Tersembunyi					
		1	2	3	4	5	6
Simpul Lapisan Masukan	0	-0,758	0,313	-0,788	-0,738	0,037	-0,695
	1	0,216	0,327	0,148	0,189	0,202	0,215
	2	0,279	-0,010	0,213	-0,177	0,246	-0,209
	3	-0,256	0,215	0,201	0,437	-0,216	0,005
	4	0,284	-0,256	-0,089	-0,451	0,174	0,170
	5	0,091	-0,056	0,129	-0,059	0,150	0,334
	6	-0,277	0,297	-0,272	-0,115	-0,326	0,392
	7	-0,152	0,209	0,170	0,257	-0,368	0,040
	8	0,032	0,328	-0,387	0,286	-0,002	-0,309
	9	0,315	0,111	-0,184	-0,304	0,444	-0,299
	10	0,320	-0,332	-0,375	-0,010	-0,154	-0,207
	11	-0,235	0,249	-0,333	-0,053	0,082	0,291
12	0,324	0,310	0,267	0,142	-0,267	-0,210	
		7	8	9	10	11	12
Simpul Lapisan Masukan	0	0,548	0,547	0,383	-0,603	0,275	0,032
	1	0,328	0,335	0,070	0,150	-0,283	-0,297
	2	-0,268	-0,172	-0,031	0,197	0,333	-0,221
	3	0,448	0,206	-0,499	-0,039	-0,357	-0,331
	4	-0,157	0,204	-0,167	-0,331	0,198	-0,340
	5	-0,317	-0,096	-0,345	-0,216	0,229	0,345
	6	-0,260	0,054	0,301	0,329	0,266	0,075
	7	0,121	-0,340	-0,193	-0,276	-0,299	0,047
	8	-0,028	-0,358	0,029	0,259	-0,072	-0,332
	9	-0,155	0,025	-0,342	0,031	-0,173	0,330
	10	0,345	0,224	0,104	0,394	0,216	0,114
	11	0,089	0,348	-0,242	-0,335	-0,049	-0,134
12	0,052	-0,297	0,158	-0,046	0,296	0,012	

Setelah bobot lapisan tersembunyi diinisialisasi dengan teknik Nguyen-Widrow, berikutnya melakukan inisialisasi bobot lapisan keluaran. Bobot lapisan keluaran (w_k) beserta biasnya (w_0), diinisialisasi dengan bilangan acak kecil dari

minus 0,5 sampai 0,5. Berikut ini adalah hasil inisialisasi untuk bobot lapisan keluaran.

Tabel 4.6 Inisialisasi Bobot Lapisan Keluaran

	<i>Output</i>
0	0.473
1	0.149
2	-0.300
3	-0.046
4	-0.068
5	0.325
6	-0.417
7	-0.367
8	-0.327
9	-0.109
10	0.331
11	0.303
12	-0.440

4.3.4.2 Inisialisasi Variabel *Learning Rate* dan Momentum

Setelah semua bobot diinisialisasi, langkah selanjutnya yaitu inisialisasi variabel *learning rate* dan momentum. Inisialisasi variabel *learning rate* (laju pembelajaran) dan momentum diperlukan untuk perubahan bobot. Variabel *learning rate* dan momentum ditentukan dengan rentang 0 sampai 1 (Siang, 2009). Nilai variabel *learning rate* adalah 0,3 sedangkan momentumnya 0,6.

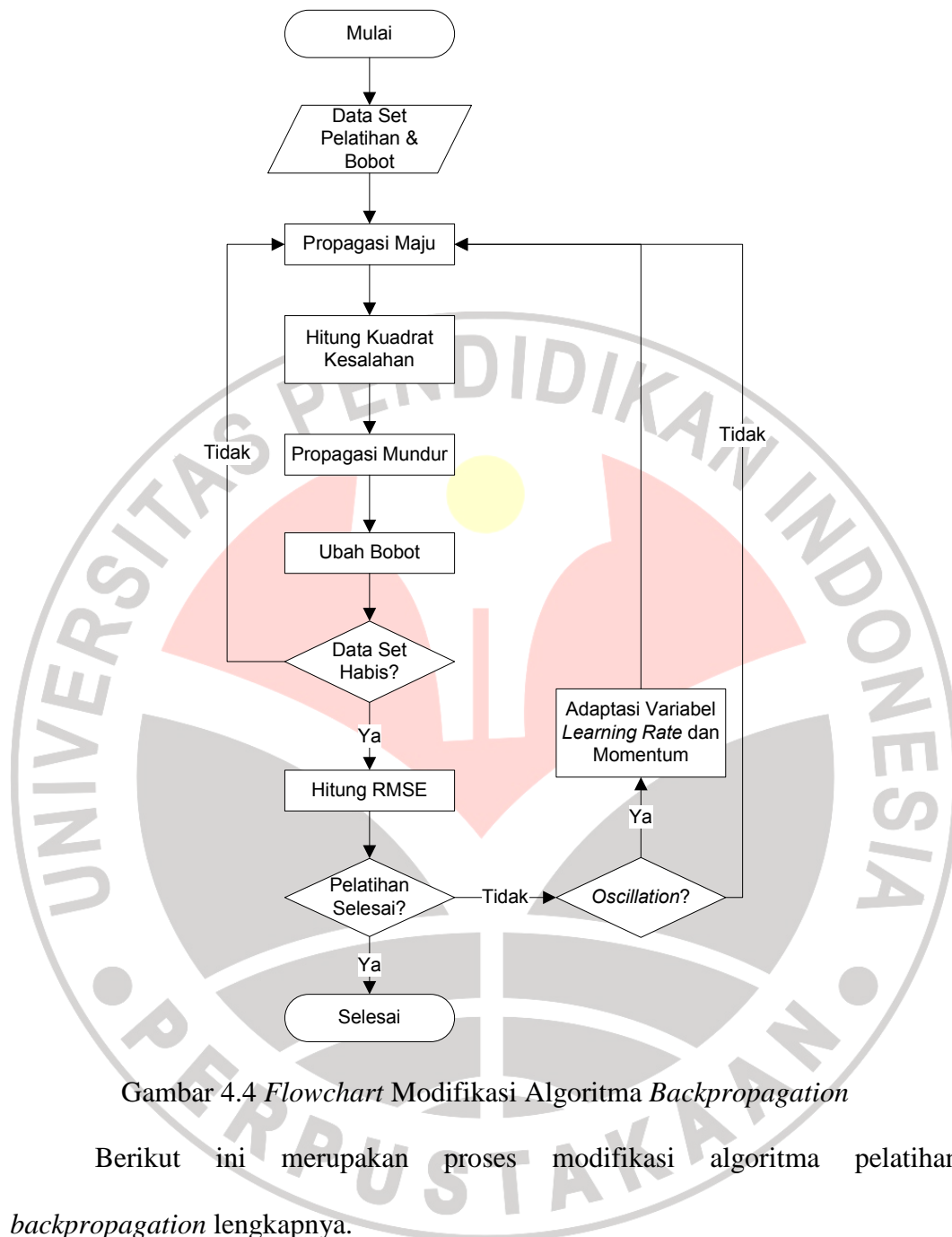
4.3.4.3 Inisialisasi Epoch Maksimum dan Batas Nilai Toleransi

Pada studi kasus awal epoch maksimum jaringan ditentukan sebanyak 400, dengan batas nilai toleransi kesalahan 0,01. Nilai toleransi tersebut ditentukan dengan asumsi selisih nilai kesalahan yang diharapkan antara keluaran jaringan dengan target adalah 0,01. Inisialisasi epoch maksimum dan batas nilai toleransi bersifat fleksibel karena pada saat studi kasus akan digunakan nilai inisialisasi yang berbeda-beda.

4.3.5 Modifikasi Algoritma Pelatihan *Backpropagation*

Setelah tahapan pada poin 4.3.1 sampai poin 4.3.4 dilakukan, tahapan selanjutnya yaitu proses pelatihan untuk mendapatkan bobot dengan kesalahan minimum. Algoritma yang digunakan yaitu *Backpropagation* dengan terlebih dahulu dilakukan modifikasi. Algoritma *Backpropagation* standar yang telah dipaparkan pada Bab II poin 2.6.3 dimodifikasi dengan menambahkan perhitungan kesalahan dengan RMSE serta algoritma optimalisasi *adaptive learning rate* dan momentum.

Dalam tahap modifikasi *backpropagation*, terdapat beberapa proses yang perlu dilakukan pengulangan. Untuk lebih jelasnya, proses-prosesnya dapat digambarkan dengan *flowchart* seperti terlihat pada gambar 4.4 berikut.



Gambar 4.4 *Flowchart* Modifikasi Algoritma *Backpropagation*

Berikut ini merupakan proses modifikasi algoritma pelatihan *backpropagation* lengkapnya.

- a. Setelah sebelumnya dilakukan penyusunan data set pelatihan seperti yang telah dijelaskan pada poin 4.3.3, langkah awal ambil satu set data pelatihan. Sebagai contoh, diambil satu set data pelatihan pada tabel 4.2 yang disajikan pada tabel 4.7 berikut.

Tabel 4.7 Data Pelatihan

Simpul Masukan	1	0,414
	2	0,286
	3	0,379
	4	0,444
	5	0,359
	6	0,386
	7	0,595
	8	0,199
	9	0,447
	10	0,379
	11	0,611
	12	0,765
Target	0,466	

Sedangkan untuk bobot yang digunakan, diambil dari tabel 4.5. Untuk menghitung nilai keluaran pada simpul tersembunyi pertama, maka bobot yang diambil yaitu pada kolom nomor 1, seperti yang terlihat pada tabel 4.8 berikut.

Tabel 4.8 Bobot Simpul Tersembunyi Pertama

	1
0	-0,758
1	0,216
2	0,279
3	-0,256
4	0,284
5	0,091
6	-0,277
7	-0,152
8	0,032
9	0,315
10	0,320
11	-0,235
12	0,324

b. Tahap kedua melakukan fase propagasi maju, dengan menghitung semua keluaran di simpul tersembunyi dan simpul keluaran.

1) Hitung semua keluaran di simpul tersembunyi z_j ($j = 1, 2, \dots, 12$).

$$z_{\text{net}_j} = v_{j0} + \sum_{i=1}^{12} x_i v_{ji} \dots\dots\dots(4.7)$$

$$z_j = f(z_{\text{net}_j}) = \frac{1}{1 + e^{-z_{\text{net}_j}}} \dots\dots\dots(4.8)$$

Dengan menggunakan data masukan pada tabel 4.7 dan bobot pada tabel 4.8, maka keluaran di simpul tersembunyi pertama yaitu:

$$\begin{aligned} z_{\text{net}_1} &= -0,758 + (0,414*0,216) + (0,286*0,279) + (0,379*-0,256) + \\ &\quad (0,444*0,284) + (0,359*0,091) + (0,386*-0,277) + \\ &\quad (0,595*-0,152) + (0,199*0,032) + (0,447*0,315) + \\ &\quad (0,379*0,32) + (0,611*-0,235) + (0,765*0,324) \\ &= -0,758 \end{aligned}$$

$$z_1 = \frac{1}{1 + e^{-(-0,758)}} = 0,413$$

Dengan cara yang sama, maka nilai keluaran tersembunyi untuk simpul yang lainnya adalah:

$$\begin{aligned} z_2 &= 0,733; z_3 = 0,292; z_4 = 0,340; z_5 = 0,471; z_6 = 0,366; z_7 = 0,672; z_8 = 0,645; \\ z_9 &= 0,471; z_{10} = 0,313; z_{11} = 0,598; z_{12} = 0,450 \end{aligned}$$

2) Hitung keluaran jaringan di simpul keluaran y

$$y_{\text{net}} = w_0 + \sum_{j=1}^{12} z_j w_j \dots\dots\dots(4.9)$$

$$y = f(y_{\text{net}}) = \frac{1}{1 + e^{-y_{\text{net}}}} \dots\dots\dots(4.10)$$

Dengan menggunakan bobot lapisan keluaran pada tabel 4.6, maka keluaran jaringan adalah:

$$\begin{aligned} y_{\text{net}} &= 0,473 + (0,413*0,149) + (0,733*0,3) + (0,292*-0,046) + \\ &\quad (0,340*-0,068) + (0,470*0,325) + (0,366*-0,417) + \\ &\quad (0,672*-0,367) + (0,645*-0,327) + (0,471*-0,109) + \\ &\quad (0,313*0,331) + (0,598*-0,303) + (0,450*-0,44) \end{aligned}$$

$$= 0,298$$

$$y = \frac{1}{1 + e^{-0,298}} = 0,574$$

c. Hitung kesalahan

Untuk data set pelatihan pertama, setelah diperoleh nilai keluaran, maka kesalahan jaringan dihitung sebagai berikut.

$$SE \text{ (Square Error)} = (\text{target} - \text{output})^2$$

$$SE = (0,466 - 0,574)^2 = 0,012$$

d. Proses selanjutnya yaitu fase propagasi mundur, dimana faktor kesalahan pada setiap simpul di lapisan keluaran dan lapisan tersembunyi dihitung dan

dilakukan perubahan bobot. Berikut ini merupakan perhitungan untuk fase propagasi mundur.

- 1) Hitung faktor δ simpul keluaran

$$\delta = (t - y) * y * (1 - y) \dots\dots\dots(4.11)$$

$$= (0,466 - 0,574) * 0,574 * (1 - 0,574)$$

$$= -0,026$$

- 2) Hitung perubahan bobot w_k dengan *learning rate* atau laju pemahaman α

($\alpha=0,3$).

$$\Delta w_k = \alpha \delta z_k \dots\dots\dots(4.12)$$

$$\Delta w_0 = 0,3 * (-0,026) * 1 = -0,0079$$

$$\Delta w_1 = 0,3 * (-0,026) * 0,413 = -0,003$$

Dengan cara yang sama, perubahan bobot lainnya yaitu:

$$\Delta w_2 = -0,06; \Delta w_3 = -0,002; \Delta w_4 = -0,003; \Delta w_5 = -0,004; \Delta w_6 = -0,003;$$

$$\Delta w_7 = -0,005; \Delta w_8 = -0,005; \Delta w_9 = -0,004; \Delta w_{10} = -0,002; \Delta w_{11} = -$$

$$0,005; \Delta w_{12} = -0,004;$$

- 3) Hitung faktor δ_{net} simpul tersembunyi

$$\delta_{net_j} = \delta w_j \dots\dots\dots(4.13)$$

Faktor δ_{net} simpul tersembunyi pertama yaitu:

$$\delta_{net_1} = -0,026 * 0,149 = -0,004$$

Setelah dilakukan perhitungan dengan cara yang sama, maka faktor δ_{net} simpul lainnya adalah:

$$\delta_{_net2} = -0,008; \quad \delta_{_net3} = 0,001; \quad \delta_{_net4} = 0,002; \quad \delta_{_net5} = -0,009;$$

$$\delta_{_net6} = 0,01; \quad \delta_{_net7} = 0,01; \quad \delta_{_net8} = 0,009; \quad \delta_{_net9} = 0,003; \quad \delta_{_net10} = -$$

$$0,009; \quad \delta_{_net11} = -0,008; \quad \delta_{_net12} = 0,012;$$

4) Selanjutnya menghitung faktor δ simpul tersembunyi.

$$\delta_j = \delta_{_net_j} z_j (1 - z_j) \dots\dots\dots(4.14)$$

$$\delta_1 = -0,004 * 0,413 * (1 - 0,413) = -0,001$$

Dengan cara yang sama, dihitung faktor δ lainnya dan diperoleh hasil sebagai berikut.

$$\delta_2 = -0,002; \quad \delta_3 = 0,0003; \quad \delta_4 = 0,0004; \quad \delta_5 = -0,002; \quad \delta_6 = 0,003; \quad \delta_7 = 0,002;$$

$$\delta_8 = 0,002; \quad \delta_9 = 0,0007; \quad \delta_{10} = -0,002; \quad \delta_{11} = -0,002; \quad \delta_{12} = 0,003$$

5) Hitung suku perubahan bobot v_{ji}

$$\Delta v_{ji} = \alpha \delta_j x_i \dots\dots\dots(4.15)$$

$$\Delta v_{10} = 0,3 * (-0,001) * 1 = -0,0003$$

$$\Delta v_{11} = 0,3 * (-0,001) * 0,414 = -0,0001$$

$$\Delta v_{20} = 0,3 * (-0,002) * 1 = -0,0005$$

$$\Delta v_{21} = 0,3 * (-0,002) * 0,414 = -0,0002$$

Perhitungan perubahan bobot keseluruhan dapat dilihat pada tabel 4.9 berikut ini.

Tabel 4.9 Perubahan Bobot Lapisan Tersembunyi

		Simpul Lapisan Tersembunyi					
		1	2	3	4	5	6
Simpul Lapisan Masukan	0	-0,0003	-0,0005	0,0001	0,0001	-0,0006	0,0008
	1	-0,0001	-0,0002	0,0000	0,0000	-0,0003	0,0003
	2	-0,0001	-0,0001	0,0000	0,0000	-0,0002	0,0002
	3	-0,0001	-0,0002	0,0000	0,0000	-0,0002	0,0003
	4	-0,0001	-0,0002	0,0000	0,0001	-0,0003	0,0003
	5	-0,0001	-0,0002	0,0000	0,0000	-0,0002	0,0003
	6	-0,0001	-0,0002	0,0000	0,0000	-0,0002	0,0003
	7	-0,0002	-0,0003	0,0000	0,0001	-0,0004	0,0005
	8	-0,0001	-0,0001	0,0000	0,0000	-0,0001	0,0002
	9	-0,0001	-0,0002	0,0000	0,0001	-0,0003	0,0003
	10	-0,0001	-0,0002	0,0000	0,0000	-0,0002	0,0003
	11	-0,0002	-0,0003	0,0000	0,0001	-0,0004	0,0005
	12	-0,0002	-0,0004	0,0001	0,0001	-0,0005	0,0006
		7	8	9	10	11	12
Simpul Lapisan Masukan	0	0,0006	0,0006	0,0002	-0,0006	-0,0006	0,0009
	1	0,0003	0,0002	0,0001	-0,0002	-0,0002	0,0004
	2	0,0002	0,0002	0,0001	-0,0002	-0,0002	0,0002
	3	0,0002	0,0002	0,0001	-0,0002	-0,0002	0,0003
	4	0,0003	0,0003	0,0001	-0,0003	-0,0003	0,0004
	5	0,0002	0,0002	0,0001	-0,0002	-0,0002	0,0003
	6	0,0002	0,0002	0,0001	-0,0002	-0,0002	0,0003
	7	0,0004	0,0004	0,0001	-0,0003	-0,0003	0,0005
	8	0,0001	0,0001	0,0000	-0,0001	-0,0001	0,0002
	9	0,0003	0,0003	0,0001	-0,0003	-0,0003	0,0004
	10	0,0002	0,0002	0,0001	-0,0002	-0,0002	0,0003
	11	0,0004	0,0004	0,0001	-0,0003	-0,0004	0,0005
	12	0,0005	0,0005	0,0002	-0,0004	-0,0004	0,0007

- e. Selanjutnya menghitung perubahan bobot lapisan tersembunyi dan lapisan keluaran dengan momentum $\mu = 0,5$. Perhitungan perubahan bobot lapisan keluaran disajikan berikut ini.

$$w_{k(t+1)} = w_{k(t)} + \Delta w_k + \mu(w_{k(t)} - w_{k(t-1)}) \dots\dots\dots(4.16)$$

$$w_{0(t+1)} = 0,473 + (-0,0079) + 0,5 (0,473 - 0) = 0,461$$

$$w_{1(t+1)} = 0,149 + (-0,003) + 0,5 (0,149 - 0) = 0,144$$

Setelah dilakukan perhitungan terhadap bobot yang lain, maka diperoleh:

$$w_2=0,292; w_3= -0,05; w_4= -0,07; w_5=0,320; w_6= -0,421;$$

$$w_7= -0,375; w_8= -0,334; w_9= -0,115; w_{10}=0,328; w_{11}=0,296;$$

$$w_{12}= -0,445$$

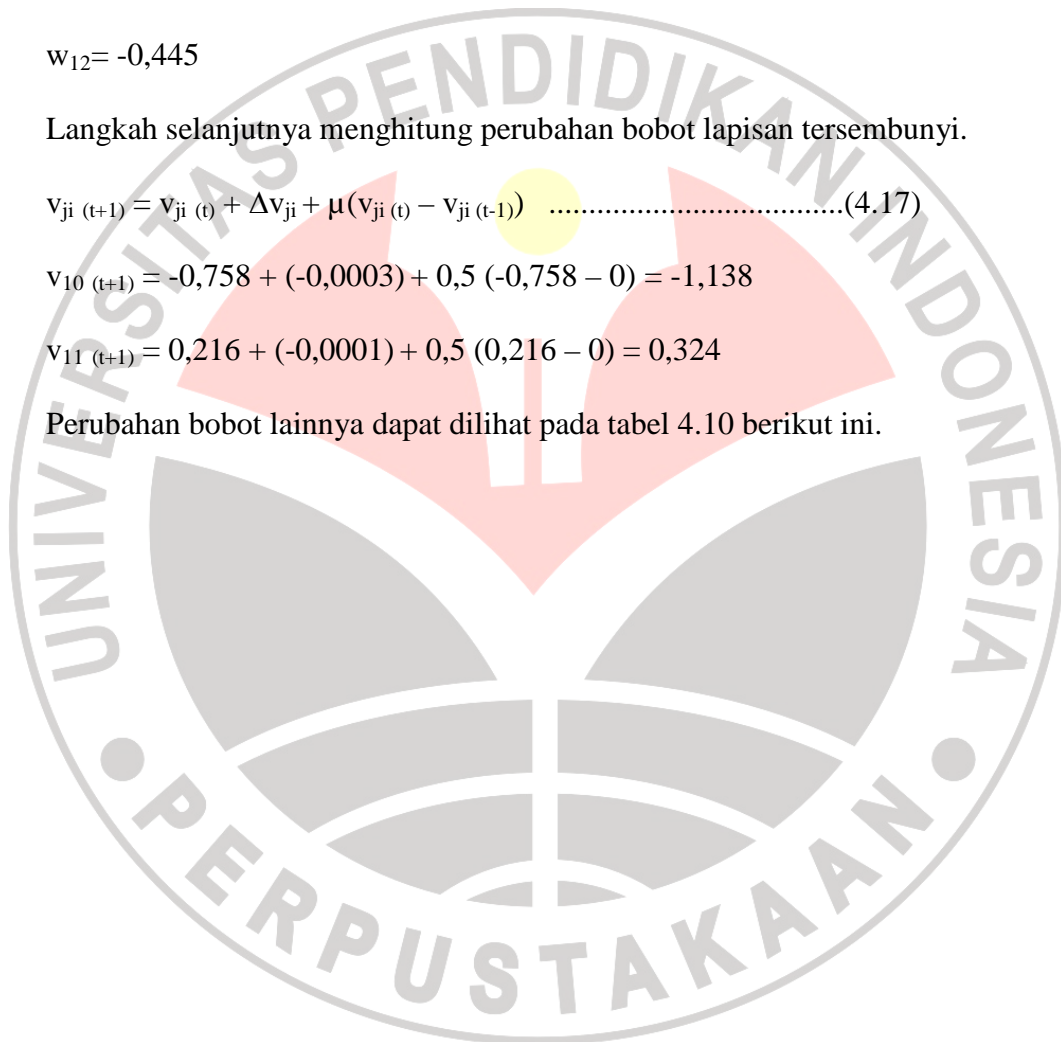
Langkah selanjutnya menghitung perubahan bobot lapisan tersembunyi.

$$v_{ji(t+1)} = v_{ji(t)} + \Delta v_{ji} + \mu(v_{ji(t)} - v_{ji(t-1)}) \dots\dots\dots(4.17)$$

$$v_{10(t+1)} = -0,758 + (-0,0003) + 0,5 (-0,758 - 0) = -1,138$$

$$v_{11(t+1)} = 0,216 + (-0,0001) + 0,5 (0,216 - 0) = 0,324$$

Perubahan bobot lainnya dapat dilihat pada tabel 4.10 berikut ini.



Tabel 4.10 Bobot Baru Lapisan Tersembunyi

		Simpul Lapisan Tersembunyi					
		1	2	3	4	5	6
Simpul Lapisan Masukan	0	-1,1379	-1,1381	-1,1375	-1,1375	-1,1382	-1,1368
	1	0,3245	0,3244	0,3246	0,3246	0,3243	0,3249
	2	0,4184	0,4184	0,4185	0,4185	0,4183	0,4187
	3	-0,3847	-0,3848	-0,3846	-0,3846	-0,3848	-0,3843
	4	0,4262	0,4261	0,4263	0,4264	0,4260	0,4266
	5	0,1364	0,1363	0,1365	0,1365	0,1363	0,1368
	6	-0,4152	-0,4152	-0,4150	-0,4150	-0,4153	-0,4148
	7	-0,2286	-0,2287	-0,2284	-0,2284	-0,2288	-0,2280
	8	0,0482	0,0482	0,0483	0,0483	0,0482	0,0485
	9	0,4716	0,4715	0,4718	0,4718	0,4715	0,4721
	10	0,4793	0,4792	0,4794	0,4794	0,4792	0,4797
	11	-0,3533	-0,3534	-0,3531	-0,3530	-0,3535	-0,3526
12	0,4850	0,4849	0,4853	0,4853	0,4848	0,4858	
		7	8	9	10	11	12
Simpul Lapisan Masukan	0	-1,1370	-1,1370	-1,1374	-1,1382	-1,1382	-1,1367
	1	0,3249	0,3248	0,3247	0,3244	0,3244	0,3250
	2	0,4187	0,4187	0,4186	0,4183	0,4183	0,4187
	3	-0,3844	-0,3844	-0,3845	-0,3848	-0,3848	-0,3843
	4	0,4266	0,4266	0,4264	0,4260	0,4260	0,4267
	5	0,1367	0,1367	0,1366	0,1363	0,1363	0,1368
	6	-0,4148	-0,4148	-0,4150	-0,4153	-0,4153	-0,4147
	7	-0,2281	-0,2281	-0,2283	-0,2288	-0,2288	-0,2279
	8	0,0484	0,0484	0,0483	0,0482	0,0482	0,0485
	9	0,4720	0,4720	0,4718	0,4715	0,4715	0,4721
	10	0,4796	0,4796	0,4795	0,4792	0,4792	0,4797
	11	-0,3527	-0,3527	-0,3530	-0,3534	-0,3535	-0,3526
12	0,4857	0,4857	0,4854	0,4848	0,4848	0,4859	

- f. Setelah bobot untuk data set pertama diperbaharui, maka proses kembali ke fase propagasi maju dengan pola data baru, yaitu data set kedua. Untuk data set kedua dilakukan proses seperti data set pertama, dari propagasi maju sampai

bobot kembali diperbaharui. Proses ini berulang terus sampai data set terakhir, yaitu data set ke-70.

- g. Setiap data set pelatihan dihitung kuadrat kesalahannya atau SE (seperti yang telah dibahas pada fase propagasi maju) dan diakumulasikan sampai data set terakhir. Setelah itu dibagi dengan jumlah data set pelatihannya, yaitu sebanyak 70 dan diakarkan. Atau dengan kata lain, RMSE-nya dihitung dengan rumus:

$$\text{RMSE} = \sqrt{\frac{1}{n} \text{SE}} \dots\dots\dots(4.18)$$

Setelah RMSE dihitung, berarti satu kali pelatihan (satu kali epoch) selesai.

- h. Setelah satu kali epoch berakhir, tahap berikutnya menghitung selisih RMSE saat ini dengan RMSE sebelumnya. Hal ini diperlukan untuk melakukan adaptasi nilai variabel *learning rate* (α) dan momentum (μ) yang didasarkan pada algoritma yang telah dipaparkan pada Bab II poin 2.6.6.2. Adaptasi α dan μ dimaksudkan untuk memperbaharui nilai variabelnya yang ditandai dengan batas *oscillation* (osc). Osc diinisialisasi dengan 0. Apabila selisih RMSE kurang dari 0, maka osc bertambah 1. Sebaliknya, apabila selisih RMSE lebih besar sama dengan 0, maka osc berkurang satu. Setelah osc bernilai 5 atau -5, maka α dan μ diperbaharui dengan rumus:

$$\alpha_{(t+1)} = \alpha(t) * (\text{maksimum_epoch} - \text{osc}) / \text{maksimum epoch}; \dots\dots\dots(4.19)$$

$$\mu_{(t+1)} = \mu(t) * ((1 - \alpha_{(t)}) + \alpha_{(t+1)}); \dots\dots\dots(4.20)$$

$$\text{osc}=0;$$

- i. Jika nilai RMSE kurang dari batas nilai toleransi yang telah ditetapkan sebelumnya, yaitu 0,01, atau epoch maksimum dicapai, maka pelatihan selesai. Jika tidak, maka epoch dilanjutkan. Epoch dilanjutkan dengan melakukan proses pada poin a sampai e. Data set yang digunakan yaitu data set pelatihan seperti sebelumnya yang dimulai dari data set pertama sampai terakhir.
- j. Setelah pelatihan berakhir, bobot yang terakhir diperbaharui akan digunakan untuk proses pengujian jaringan.

4.3.6 Pengujian Jaringan

Untuk mengukur kehandalan jaringan yang telah dilatih, maka diperlukan pengujian. Pada proses pengujian, arsitektur jaringan sama dengan arsitektur pada saat pelatihan. Bobot yang digunakan pun merupakan bobot hasil pelatihan. Karena pada saat pengujian hanya diperlukan nilai keluaran, maka hanya dilakukan fase propagasi maju seperti yang telah dijelaskan pada tahap modifikasi algoritma *backpropagation* poin 4.3.5.

Setelah nilai keluaran didapat, proses dilanjutkan dengan de-transformasi data. De-transformasi data merupakan proses transformasi kembali nilai keluaran yang diperoleh jaringan ke nilai curah hujan yang sebenarnya (dalam mm).

Berdasarkan rumus transformasi yang digunakan, yaitu:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Maka untuk mendapatkan nilai yang sebenarnya (x), adalah:

$$x = x' (x_{\max} - x_{\min}) + x_{\min} \dots\dots\dots(4.21)$$

Proses dilanjutkan dengan menghitung persen kesalahannya dengan menggunakan rumus MAPE dengan rumus sebagai berikut.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_t - y'_t}{y_t} \times 100\% \right| \dots\dots\dots(4.22)$$

dengan,

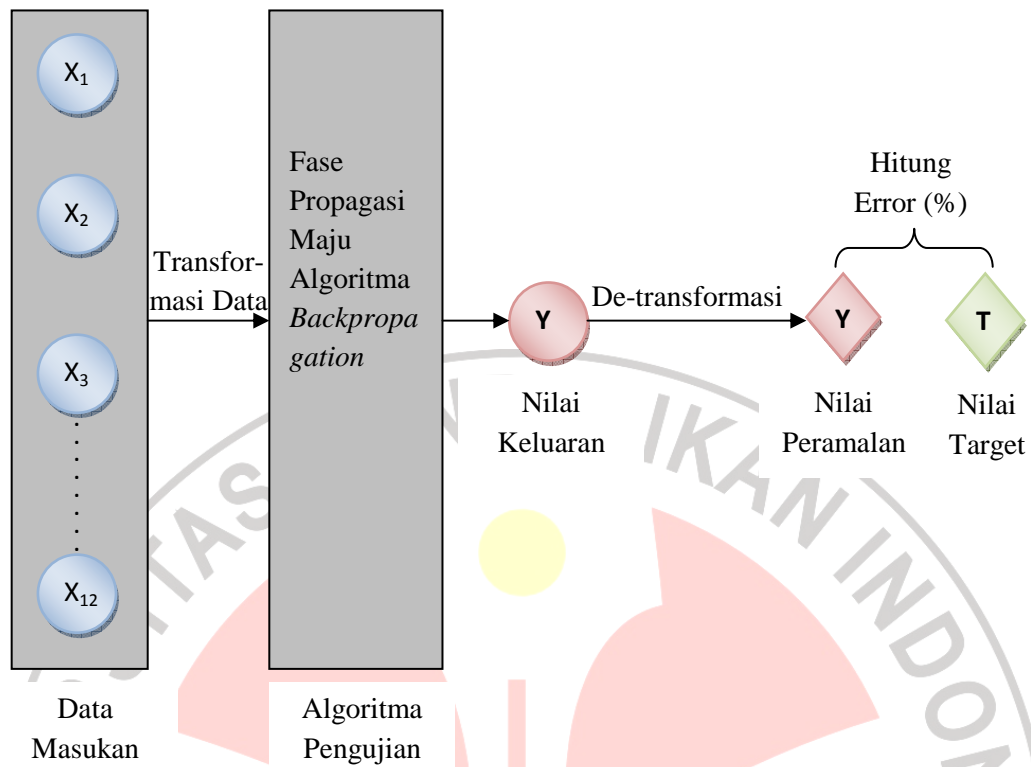
y_t = nilai aktual pada waktu t

y'_t = nilai ramalan pada waktu t

n = banyaknya ramalan

Proses pengujian dilakukan dua tahap. Pertama data yang digunakan untuk pengujian adalah data set pelatihan dan kedua data set pengujian. Data set pelatihan diuji dengan tujuan untuk melihat kemampuan jaringan mengenali pola yang diberikan. Sedangkan data set pengujian digunakan untuk mengetahui kemampuan jaringan dalam meramalkan jumlah curah hujan.

Agar lebih jelas, proses pengujian jaringan untuk mendapatkan nilai peramalan curah hujan secara sederhana dapat dilihat pada gambar 4.5 berikut.



Gambar 4.5 Proses Pengujian Jaringan

4.3.7 Analisis Sensitifitas

Setelah semua proses pelatihan dan pengujian selesai dilakukan, kemudian dilakukan analisis sensitifitas. Analisis sensitifitas dilakukan dengan cara mengubah parameter-parameter jaringan. Hal ini dilakukan untuk menentukan arsitektur jaringan terbaik untuk peramalan. Parameter-parameter yang diubah yaitu: jumlah simpul layar tersembunyi, inialisasi variabel *learning rate*, momentum, epoch maksimum, dan batas nilai toleransi.

4.3.8 Pemilihan Jaringan Optimum untuk Peramalan

Langkah-langkah pemilihan jaringan yang optimum sebagai berikut.

- a. Dari hasil pelatihan dan pengujian dengan berbagai kasus data inisialisasi yang berbeda-beda, dipilih jaringan dengan nilai MAPE data pengujian terendah.
- b. Perlu diperhatikan juga nilai MAPE data pelatihan. Bila nilai MAPE data pelatihan lebih besar dari MAPE data pengujian, berarti jaringan belum dapat mengenali pola. Tentulah jaringan tersebut tidak dipilih sebagai jaringan optimum.
- c. Jadi, jaringan optimum yang dipilih adalah jaringan dengan nilai MAPE data pelatihan kurang dari nilai MAPE data pengujian dan memiliki nilai MAPE data pengujian terendah.

4.4 Algoritma Peramalan Curah Hujan

Berdasarkan proses pengembangan sistem curah hujan yang telah dijelaskan pada poin 4.3, maka algoritma pelatihan, pengujian, dan peramalannya disajikan berikut.

Algoritma Pelatihan

1. Lakukan *data preprocessing* dengan proses *data cleansing*, transformasi data, dan pembagian data.
2. Tentukan arsitektur jaringan (jumlah simpul masukan, tersembunyi, dan keluaran)
3. Inisialisasi variabel *learning rate* dan momentum
4. Tentukan epoch maksimal dan nilai toleransi
5. Tentukan frekuensi maksimal dan minimal osilasi (untuk teknik *adaptive learning rate* dan momentum)
6. Susun data set pelatihan ke dalam matriks x
7. Inisialisasi bobot dengan teknik Nguyen-Widrow
 - a) Inisialisasi matriks bobot lapisan tersembunyi (v_{ji}) dan lapisan keluaran (w_k) dengan bilangan acak pada interval [-0,5 sampai 0,5]
 - b) Hitung $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2}$ untuk v_{ji}

$$\beta = (0,7 \sqrt[n]{p})$$
 ; dengan $n = \text{jmlh simpul masukan}$ dan $p = \text{jmlh simpul tersembunyi}$

$$v_{ji} = \frac{\beta v_{ji}(\text{lama})}{\|v_j\|}$$

$$v_{j0} (\text{bias}) = \text{bilangan acak antara } -\beta \text{ dan } \beta$$
8. Selama epoch $\leq \text{max_epoch}$ dan RMSE $>$ toleransi lakukan langkah 9 sampai terakhir
9. Untuk setiap data set pelatihan, lakukan langkah 10 – 15

10. Hitung semua keluaran di simpul tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_{\text{net}_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_{\text{net}_j}) = \frac{1}{1 + e^{-z_{\text{net}_j}}}$$

11. Hitung keluaran jaringan di simpul keluaran y .

$$y_{\text{net}} = w_0 + \sum_{j=1}^p z_j w_j$$

$$y = f(y_{\text{net}}) = \frac{1}{1 + e^{-y_{\text{net}}}}$$

12. Hitung nilai kesalahan

$$SE(\text{Error}) = (\text{target} - y)^2$$

13. Hitung faktor δ unit keluaran berdasarkan kesalahan di simpul keluaran y .

$$\delta = (t - y) f'(y_{\text{net}}) = (t - y) y(1 - y)$$

dengan:

t = target keluaran

δ merupakan simpul kesalahan yang akan dipakai dalam perubahan bobot lapisan di bawahnya.

Hitung perubahan bobot w_{kj} dengan laju pemahaman α .

$$\Delta w_k = \alpha \delta z_k \text{ dengan } k = 1, 2, \dots, m ; j = 0, 1, \dots, p$$

14. Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$\delta_{\text{net}_j} = \sum_{j=1}^m \delta w_j$$

Faktor δ unit tersembunyi.

$$\delta_j = \delta_{\text{net}_j} f'(y_{\text{net}_j}) = \delta_{\text{net}_j} z_j (1 - z_j)$$

Hitung suku perubahan bobot v_{ji} .

$$\Delta v_{ji} = \alpha \delta_j x_i \text{ dengan } j = 1, 2, \dots, p ; i = 0, 1, \dots, n$$

15. Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran, dengan momentum μ , yaitu:

$$w_{k(t+1)} = w_{k(t)} + \Delta w_k + \mu(w_{k(t)} - w_{k(t-1)}) ; (k=1, 2, \dots, m ; j=0, 1, \dots, p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi, yaitu:

$$v_{ji(t+1)} = v_{ji(t)} + \Delta v_{ji} + \mu(v_{ji(t)} - v_{ji(t-1)}) ; (j=1, 2, \dots, p ; i=0, 1, \dots, n)$$

16. Hitung rata-rata kesalahan

$$\text{RMSE (Root Mean Square Error)} = \sqrt{(\sum \text{SE} / \text{byk_data_set})}$$

17. Lakukan teknik optimasi

$$\Delta E := \text{RMSE}_{(\text{sebelumnya})} - \text{RMSE}_{(\text{sekarang})}$$

if $\Delta E < 0$ then

oscilasi+1

else if $\Delta E \geq 0$ then

oscilasi-1

end if

if (oscilasi > max_freq) or (oscilasi < min_freq) then

$$\alpha = (\alpha_{(\text{sebelumnya})} * (\text{max_epoch} - \text{oscilasi})) / \text{max_epoch};$$

$$\mu = \mu_{(\text{sebelumnya})} * ((1,0 - \alpha_{(\text{sebelumnya})}) + \alpha);$$

oscilasi = 0;

end if

Algoritma Pengujian

1. Gunakan jaringan yang telah dilatih
2. Pilih data set pengujian
3. Susun data set pengujian ke dalam matriks x
4. Untuk setiap data set pengujian lakukan langkah 4 – 7
5. Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}}$$

6. Hitung semua keluaran jaringan di unit keluaran y.

$$y_net = w_0 + \sum_{j=1}^p z_j w_j$$

$$y = f(y_net) = \frac{1}{1 + e^{-y_net}}$$

7. De-transformasi nilai keluaran

$$x = x' (x_{\max} - x_{\min}) + x_{\min}$$

8. Hitung nilai kesalahan

$$E (\text{Error}) = X_{(\text{jaringan})} - X_{(\text{aktual})}$$

$$\text{APE (Absolute Percentage Error)} = (||E||/\text{target}) \times 100\%$$

9. Hitung rata-rata kesalahan

$$\text{MAPE (Mean Absolute Percentage Error)} = \Sigma \text{APE} / \text{byk_data_set}$$

Algoritma Peramalan

1. Gunakan jaringan optimum sesuai dengan bulan yang akan diramal.
2. Baca data masukan
3. Transformasi ke dalam rentang 0-1, dimana nilai maksimum dan minimum diambil dari data curah hujan yang telah dilakukan *preprocessing*.
4. Susun data masukan ke dalam matriks x
5. Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$).

$$z_net_j = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

$$z_j = f(z_net_j) = \frac{1}{1 + e^{-z_net_j}}$$

6. Hitung semua keluaran jaringan di unit keluaran y.

$$y_net = w_0 + \sum_{j=1}^p z_j w_j$$

$$y = f(y_net) = \frac{1}{1 + e^{-y_net}}$$

7. De-transformasi nilai keluaran

$$x = x' (x_{\max} - x_{\min}) + x_{\min}$$

8. Nilai hasil transformasi merupakan nilai peramalan.

4.5 Analisis Kebutuhan Perangkat Lunak

4.5.1 Batasan Perangkat Lunak

Sistem Peramalan Curah Hujan (PERAHU) adalah sistem yang berdiri sendiri, tidak terhubung atau tergantung dengan sistem lainnya. PERAHU menggunakan metode jaringan saraf tiruan *backpropagation*. Adapun batasan-batasan dalam pengembangan perangkat lunak peramalan curah hujan adalah sebagai berikut.

- a. Data curah hujan yang akan diunggah ke sistem untuk dilakukan tahap *preprocessing*, disusun dalam bentuk matriks satu kolom dan banyaknya baris disesuaikan dengan banyaknya data pada kelompok bulan yang sama. Satuan curah hujan dalam millimeter (mm) dan disimpan dalam file text (.txt). Bila data curah hujan mengandung koma (,) maka *engineer* terlebih dahulu harus mengubahnya ke titik(.).
- b. Sebelum diunggah ke sistem, penanganan data yang hilang terlebih dahulu harus dilakukan oleh *engineer*.
- c. Pada tahap pelatihan data, banyaknya *node hidden* (simpul tersembunyi), inisialisasi nilai variabel *learning rate* dan momentum, epoch maksimal, dan toleransi *error* dimasukkan oleh *engineer*.
- d. *Engineer* hanya dapat melakukan proses pelatihan setelah melalui tahap *data preprocessing*. Dan hanya dapat melakukan pengujian dan peramalan setelah pelatihan selesai.

4.5.2 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak meliputi analisis proses bisnis, analisis *input* dan *ouput*, analisis pengguna, serta analisis kebutuhan informasi.

4.5.2.1 Analisis Proses Bisnis

Pada sistem Peramalan Curah Hujan (PERAHU), sebelum dapat melakukan peramalan curah hujan, tahap pertama yang dilakukan yaitu *engineer* memasukkan data curah hujan bulanan beserta atributnya (nama wilayah serta bulan dan tahun awal). Data curah hujan dalam bentuk file teks (.txt) berisi data curah hujan pada kelompok bulan yang sama. Dan dituliskan dalam bentuk matriks satu kolom. Penjelasan bentuk data curah hujan lebih lanjut dapat dilihat pada bagian representasi masalah poin 4.2.

Sistem secara otomatis akan melakukan tahap *data preprocessing*. Hasilnya ditampilkan dalam bentuk data runtun waktu curah hujan sebelum dan setelah *preprocessing*. Informasi tentang banyaknya data, nilai minimum, nilai maksimum, jumlah data *outlier*, dan jumlah pembagian data pelatihan dan pengujian juga ditampilkan. Data curah hujan hasil *preprocessing* disimpan ke dalam *database*.

Data yang telah melalui tahap *preprocessing* dapat berlanjut ke tahap pelatihan. Sebelum pelatihan dimulai, *engineer* harus memasukkan data inisialisasi jumlah *node hidden*, variabel *learning rate* dan momentum, epoch maksimum, dan batas toleransi kesalahan. Pelatihan jaringan dimaksudkan agar jaringan mendapatkan bobot yang mendekati antara data masukan dengan nilai

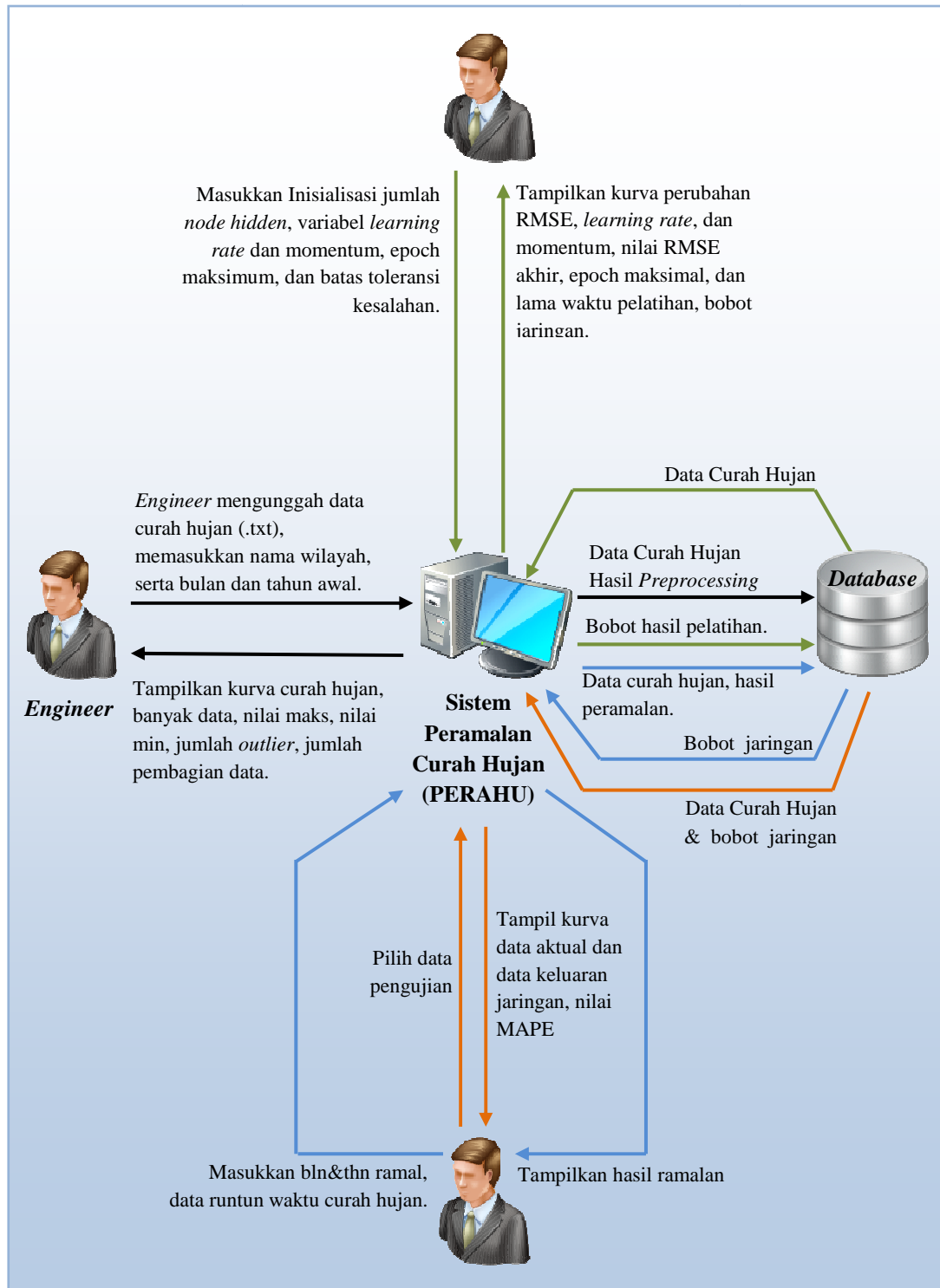
target yang diharapkan. Lamanya waktu pelatihan tergantung data inisialisasi yang dimasukkan dan perangkat keras yang digunakan. Setelah pelatihan selesai, sistem akan menampilkan kurva perubahan RMSE (*Root Mean Square Error*), *learning rate*, dan momentum. Informasi RMSE akhir, epoch maksimal yang dicapai, dan lama waktu pelatihan ditampilkan juga. *Engineer* juga dapat melihat bobot awal dan akhir pelatihan. Bobot yang diperoleh dari hasil pelatihan dapat disimpan ke dalam *database*.

Setelah dilakukan pelatihan jaringan, proses pengujian dan peramalan dapat langsung dilakukan. Pada tahap pengujian, *engineer* dapat dengan mudah memilih data pelatihan atau data pengujian yang akan dilihat. Perbandingan data sebenarnya dengan data keluaran jaringan disajikan dalam bentuk kurva. Persentase kesalahan hasil pengujian jaringan ditampilkan dalam MAPE (*Mean Absolute Percentage Error*).

Tahap terakhir yaitu peramalan. Pada tahapan ini, *engineer* diminta untuk memasukkan bulan dan tahun yang akan diramal. Perlu diperhatikan bulan yang akan diramal. Apabila pada bulan Januari maka gunakan jaringan yang telah dilatih dengan menggunakan data bulan Januari. Begitupun untuk bulan-bulan lainnya.

Kemudian data runtun waktu diisi satu persatu dalam satuan mm. Banyaknya data yang harus diisi disesuaikan dengan banyaknya simpul masukan jaringan yang telah dilatih. Setelah semua data curah hujan diisi, sistem akan menampilkan hasil peramalannya dalam bentuk angka dalam satuan mm. Data-data curah hujan yang dimasukkan dan hasil peramalannya disimpan ke dalam

database. Untuk lebih memperjelas, proses bisnis PERAHU dapat dilihat pada gambar 4.6 berikut.



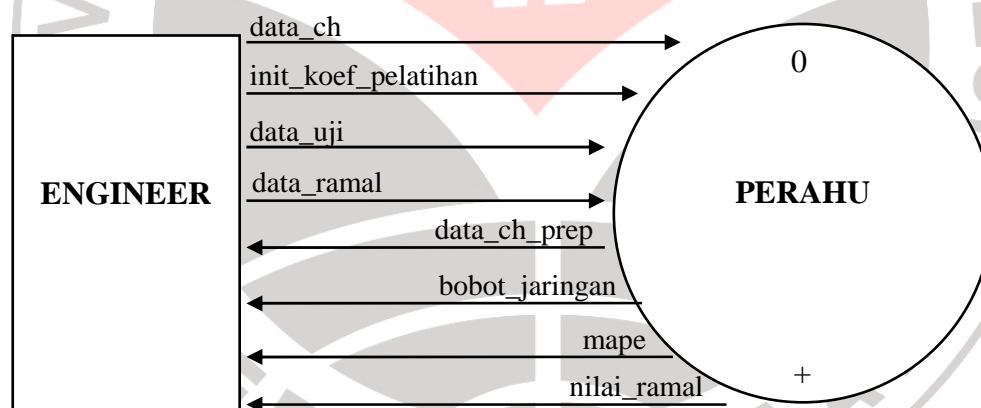
Gambar 4.6 Proses Bisnis Sistem Peramalan Curah Hujan

4.5.3 Pemodelan Kebutuhan

Dalam membangun perangkat lunak, diperlukan proses analisis kebutuhan perangkat lunak. Pemodelan kebutuhan perangkat lunak dilakukan dengan menggunakan *context diagram*, *data flow diagram*, *data dictionary*, dan *process spesification*.

4.5.3.1 Model Hubungan Luar

Hubungan atau interaksi perangkat lunak dengan pengguna dan data-data atau informasi yang berpindah dan berubah dapat dimodelkan ke dalam sebuah *Context Diagram*. *Context Diagram* untuk perangkat lunak ini dapat dilihat pada gambar 4.7 berikut ini.



Gambar 4.7 *Context Diagram* PERAHU

Secara umum, pengguna PERAHU adalah *engineer* yang mengerti tentang metode *backpropagation* dan peramalan curah hujan. Untuk dapat menjalankan PERAHU dengan baik, pertama-tama *engineer* harus memasukkan *data_ch* yang berupa data curah hujan pada bulan yang sama beserta atributnya (nama wilayah serta bulan dan tahun awal data). Data yang dimasukkan kemudian dilakukan *data*

preprocessing yang akan menghasilkan *data_ch_prep* yang merupakan data curah hujan yang siap digunakan pelatihan.

Pada tahap pelatihan, *engineer* memasukkan *init_koef_pelatihan* yang berisi data inialisasi jumlah simpul tersembunyi, variabel *learning rate* dan momentum, maksimum epoch, dan batas toleransi kesalahan. Sesuai dengan tujuan pelatihan jaringan, setelah pelatihan selesai, diperoleh *bobot_jaringan* yang merupakan nilai bobot lapisan tersembunyi dan lapisan keluaran.

Pada tahap pengujian, *engineer* hanya memilih *data_uji*. Terdapat dua pilihan data pengujian, yaitu data pengujian yang juga digunakan pada data pelatihan serta data pengujian yang belum pernah dilatihkan. Data yang dipilih kemudian diuji dan menghasilkan nilai *mape* yang merupakan nilai persentase kesalahan jaringan.

Terakhir, ketika akan meramalkan nilai curah hujan bulan tertentu, *engineer* harus memasukkan bulan dan tahun yang akan diramal, serta data runtun waktu curah hujan (*data_ramal*) yang jumlahnya sesuai dengan simpul masukan jaringan.

Model tingkah laku atau behaviour dari perangkat lunak dapat dimodelkan melalui *Data Flow Diagram* (DFD). Penjelasan lengkap mengenai DFD dari perangkat lunak yang dibangun, kamus data, spesifikasi proses, dan model data dapat dilihat pada Dokumen Teknis.

4.5.4 Desain Perangkat Lunak

Perangkat lunak PERAHU menggunakan *database* untuk menyimpan data curah hujan, data hasil pelatihan, dan data peramalan. Berikut ini merupakan desain *database* yang digunakan.

a. *Database preprocessing*

Database yang dinamakan *preprocessing*, berisi *no* yang merupakan *primary key* dan terisi secara otomatis, *nama_prep* yang merupakan nama wilayah data curah hujan, *bln_awal*, *thn_awal*, *n_max* berupa nilai maksimum data, *n_min* berupa nilai minimum data, *n_out* yang merupakan banyaknya nilai *outlier*, *n_latih* yang merupakan banyaknya data pelatihan, *n_uji* yang merupakan banyaknya data pengujian, dan *data_ch* yang berisi data curah hujan hasil *preprocessing*.

b. *Database jaringan*

Database jaringan berisi *no* yang merupakan *primary key* dan terisi secara otomatis, *no_prep* berupa *foreign key* *no* pada *database preprocessing*, *nama_proyek*, *node_input*, *node_hidden*, *node_output*, *alfa* yang merupakan variabel *learning rate*, *miu* yang merupakan variabel momentum, *max_epoch*, toleransi yang merupakan batas nilai toleransi, *epoch* yang merupakan total *epoch* yang dicapai, *wkt_latih*, *rmse*, *bobot_v1* yang merupakan nilai bobot akhir lapisan tersembunyi, *bobot_w1* yang merupakan nilai bobot akhir lapisan keluaran, *i_bobot_v1* yang merupakan bobot awal lapisan tersembunyi, dan *i_bobot_w1* yang merupakan bobot awal lapisan keluaran.

c. *Database ramalan*

Database ramalan berisi no yang merupakan *primary key* dan terisi secara otomatis, no_jar yang merupakan *foreign key* no pada *database* jaringan, bulan, tahun, data_ramal, dan hasil ramal.

4.5.5 Implementasi

Dari hasil penelitian dan desain yang telah dilakukan, dibuat sebuah implementasi dan penerapan dalam bentuk kode program. Bentuk kode program merupakan wujud nyata dari hasil penelitian, berupa sistem aplikasi yang dapat digunakan. Kode program PERAHU dibagi ke dalam beberapa modul, yaitu sebagai berikut.

Tabel 4.11 Modul PERAHU

No.	Nama Modul	Keterangan
1.	layar_utama.m	Merupakan modul utama yang menampilkan antarmuka awal ketika sistem dijalankan dan sebagai wadah untuk menampilkan semua proses yang ada.
2.	layar_prep.m	Merupakan modul untuk menampilkan layar data <i>preprocessing</i> . Layar ini merupakan layar yang pertama kali dimunculkan pada layar_utama.m
3.	getfile.m	Modul ini digunakan untuk mendapatkan dan menampilkan alamat data curah hujan.
4.	preprocessing.m	Modul ini digunakan untuk melakukan proses <i>data preprocessing</i> sekaligus menyimpan hasilnya ke <i>database</i> .

		Sebelumnya, data masukan diverifikasi dengan modul ini juga.
5.	layar_hasil_prep.m	Modul ini digunakan untuk menampilkan hasil <i>preprocessing</i> .
6.	HapusPrep.m	Modul ini digunakan untuk mengonfirmasi data hasil <i>preprocessing</i> yang akan dihapus.
7.	HapusPrepNow.m	Setelah yakin data <i>preprocessing</i> akan dihapus, maka akan dihapus dengan modul ini.
8.	BukaPrep.m	Modul ini digunakan untuk membuka data hasil <i>preprocessing</i> .
9.	layar_pelatihan.m	Modul ini digunakan untuk menampilkan layar pelatihan.
10.	TampilGrafikPrep.m	Modul ini digunakan untuk menampilkan grafik data curah hujan pada layar pelatihan.
11.	ramalan_ch.m	Modul ini digunakan untuk melakukan proses pelatihan jaringan yang akan diperoleh bobot baru.
12.	TampilHasilPelatihan.m	Setelah proses pelatihan selesai, maka hasilnya ditampilkan dengan modul ini.
13.	TampilGrafik.m	Modul ini digunakan untuk menampilkan grafik hasil pelatihan. Berupa RMSE, perubahan <i>learning rate</i> , dan perubahan momentum.
14.	layar_bobot.m	Modul ini digunakan untuk menampilkan layar bobot.
15.	BobotBaru.m	Modul ini digunakan untuk memilih bobot yang akan ditampilkan, bobot awal

		atau bobot baru hasil dari pelatihan.
16.	kotak_simpan.m	Modul ini digunakan untuk menampilkan kotak dialog untuk mengisi nama proyek yang akan disimpan.
17.	Simpan.m	Setelah nama proyek diisi pada modul kotak_simpan.m, maka akan dicek terlebih dahulu. Proyek dapat disimpan apabila nama tersebut belum ada sebelumnya. Bila sudah ada, user diminta untuk mengisi nama lain.
18.	SimpanHasilPelatihan.m	Modul ini digunakan untuk menyimpan data hasil pelatihan ke <i>database</i> .
19.	layar_pengujian.m	Modul ini digunakan untuk menampilkan layar pengujian.
20.	uji_ch.m	Modul ini digunakan untuk menguji hasil pelatihan yang telah dilakukan dan menampilkannya dalam bentuk kurva.
21.	layar_peramalan.m	Modul ini digunakan untuk menampilkan layar peramalan.
22.	DataRamal.m	Merupakan modul yang digunakan untuk menampilkan banyaknya data curah hujan yang harus diisi untuk melakukan peramalan. Sebelumnya, data masukan diverifikasi dengan modul ini juga.
23.	BersihData.m	Modul ini digunakan membersihkan data peramalan.
24.	NextData.m	Modul ini digunakan untuk melanjutkan pengisian data curah hujan pada peramalan.
25.	hasil_ch.m	Modul ini digunakan untuk meramalkan

		data curah hujan berdasarkan jaringan yang dipilih dan data curah hujan yang dimasukkan. Sebelumnya, data masukan diverifikasi terlebih dahulu.
26.	SimpanRamalan.m	Modul ini digunakan untuk menyimpan data ramalan ke <i>database</i> .
27.	BukaRam.m	Bulan dan tahun data yang telah diramal dapat dibuka dengan modul ini.
28.	BukaHasilPeramalan.m	Bulan dan tahun yang dipilih dengan modul BukaRam.m ditampilkan dengan modul ini.
29.	layar_hsl_ramal.m	Hasil peramalan yang akan dibuka ditampilkan dengan modul ini.
30.	proyek_baru.m	Modul ini digunakan untuk membuat proyek baru, dengan tampilan seperti pertama kali sistem dibuka.
31.	buka_proyek.m	Modul ini digunakan untuk menampilkan dan memilih nama proyek hasil pelatihan
32.	BukaHasilPelatihan.m	Nama proyek yang telah dipilih pada modul buka_proyek.m dibuka dengan modul ini.
33.	hapus_proyek.m	Modul ini digunakan untuk menampilkan nama proyek yang akan akan dihapus.
34.	HapusProyek.m	Sebelum proyek dihapus, dikonfirmasi terlebih dahulu di modul ini.
35.	HapusProyekNow.m	Setelah yakin akan dihapus, dengan modul ini, proyek yang telah dipilih sebelumnya akan dihapus dari <i>database</i> .
36.	koneksi.m	Modul ini digunakan untuk membuka koneksi dengan <i>database</i> .

4.6 Studi Kasus dan Hasil Penelitian

4.6.1 Studi Kasus

Studi kasus yang dilakukan yaitu sebanyak 36 studi kasus dengan skenario sebagai berikut.

Tabel 4.12 Skenario Studi Kasus

No.	Data Bulan	Simpul Tersembunyi	Learning Rate	Momentum	Epoch Maksimum	Batas Toleransi
1.	Januari	36	0,3	0,6	400	0.01
2.	Januari	36	0,3	0,6	350	0.01
3.	Januari	36	0,3	0,6	600	0,064
4.	Februari	36	0,3	0,6	400	0.01
5.	Februari	36	0,3	0,6	400	0.0104
6.	Februari	36	0,3	0,6	1000	0,01
7.	Maret	36	0,3	0,6	400	0.01
8.	Maret	36	0,3	0,6	400	0.07
9.	Maret	36	0,3	0,6	500	0,01
10.	April	36	0,3	0,6	400	0.01
11.	April	36	0,3	0,6	500	0.08
12.	April	36	0,3	0,6	500	0,01
13.	Mei	36	0,3	0,6	400	0.01
14.	Mei	36	0,3	0,6	500	0.085
15.	Mei	36	0,3	0,6	500	0,01
16.	Juni	36	0,3	0,6	400	0.01
17.	Juni	36	0,3	0,6	500	0.099
18.	Juni	36	0,3	0,6	500	0,01
19.	Juli	36	0,3	0,6	400	0.01
20.	Juli	36	0,3	0,6	400	0.0128
21.	Juli	36	0,3	0,6	500	0,01
22.	Agustus	36	0,3	0,6	400	0.01
23.	Agustus	36	0,3	0,6	450	0.01
24.	Agustus	36	0,3	0,6	500	0,01
25.	September	36	0,3	0,6	400	0.01
26.	September	36	0,3	0,6	700	0.01
27.	September	36	0,3	0,6	800	0.064
28.	Oktober	36	0,3	0,6	400	0.01
29.	Oktober	36	0,3	0,6	500	0.01
30.	Oktober	36	0,3	0,6	500	0,074

31.	November	36	0,3	0,6	400	0.01
32.	November	36	0,3	0,6	400	0.091
33.	November	36	0,3	0,6	500	0,06
34.	Desember	36	0,3	0,6	400	0.01
35.	Desember	36	0,3	0,6	400	0.093
36.	Desember	36	0,3	0,6	500	0,066

4.6.2 Hasil Penelitian

Hasil pelatihan dan pengujian untuk masing-masing studi kasus dapat dilihat pada tabel berikut.

Tabel 4.13 Hasil Studi Kasus

Studi Kasus ke-	Hasil Pelatihan			MAPE Pengujian	
	Waktu Pelatihan (jj:mm:dd)	Epoch yang Dicapai	RMSE Pelatihan	Data Latih	Data Uji
1.	00:00:21	400	0,066	27,46	43,97
2.	00:00:19	350	0,115	34,32	45,70
3.	00:00:26	483	0,0636	24,58	40,25
4.	00:00:21	400	0,115	57,36	48,87
5.	00:00:22	400	0,091	56,51	78,69
6.	00:01:00	1000	0,055	100	128
7.	00:00:22	400	0,099	73,24	32,56
8.	00:00:21	400	0,081	24,27	26,23
9.	00:01:04	500	0,062	18,56	45,31
10.	00:00:21	400	0,087	18,54	27,70
11.	00:00:23	427	0,080	16,32	32,28
12.	00:00:28	500	0,082	22,74	29,59
13.	00:00:21	400	0,077	19,83	60,48
14.	00:00:24	405	0,085	22,40	47,15
15.	00:00:27	500	0,123	24,53	63,41
16.	00:00:21	400	0,104	23,85	45,10
17.	00:00:18	345	0,099	24,44	46,43
18.	00:00:32	500	0,087	25,63	52,32
19.	00:00:22	400	0,127	18,09	29,56
20.	00:00:28	400	0,127	23,45	34,55
21.	00:00:33	500	0,103	28,67	30,57
22.	00:00:21	400	0,093	28,83	64,10
23.	00:00:24	450	0,070	22,99	60,79
24.	00:00:33	500	0,064	43,62	52,33
25.	00:00:21	400	0,111	44,67	51,78
26.	00:00:37	700	0,065	19,93	47,97

27.	00:00:39	725	0,064	16,22	53,79
28.	00:00:21	400	0,117	22,37	31,13
29.	00:00:27	500	0,097	24,01	44,64
30.	00:00:24	444	0,074	20,75	36,65
31.	00:00:21	400	0,108	43,61	28,09
32.	00:00:19	360	0,091	25,60	37,49
33.	00:00:26	500	0,089	31,45	39,96
34.	00:00:21	400	0,083	27,27	49,01
35.	00:00:20	372	0,092	18,75	37,07
36.	00:00:23	432	0,065	12,46	40,16

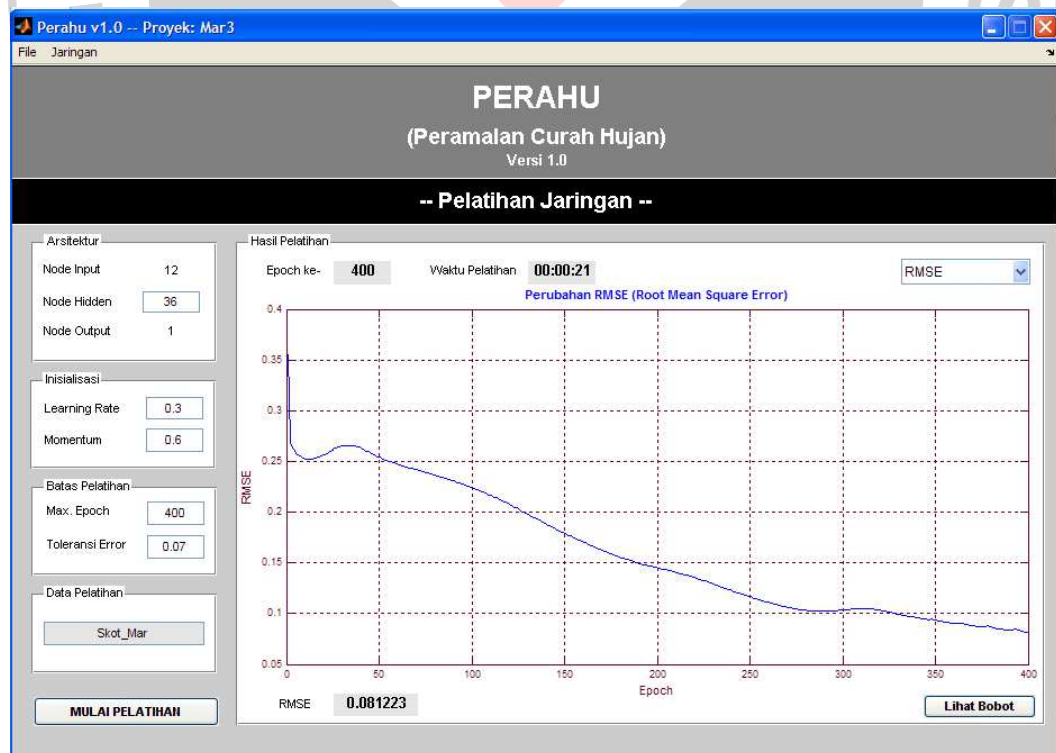
Dari seluruh studi kasus yang dilakukan, nilai MAPE optimal untuk setiap bulan dipilih berdasarkan nilai MAPE data pengujian terendah pada bulan tersebut. Dengan syarat, nilai MAPE data pelatihan harus lebih kecil dari MAPE data pengujian. pada tabel 4.13 di atas, MAPE optimum untuk setiap bulan ditandai dengan warna *background* biru. Apabila akan meramalkan bulan Desember yang akan datang, maka jaringan yang dipilih yaitu jaringan optimum pada bulan Desember. Ketika akan meramalkan bulan Januari yang akan datang, maka jaringan yang dipilih yaitu jaringan optimum pada bulan Januari. Demikian pula dengan bulan-bulan lainnya.

Bila ditinjau dari hasil studi kasus keseluruhan, MAPE optimum yang dihasilkan yaitu pada kasus ke-8 tepatnya pada bulan Maret. MAPE untuk data pelatihan yaitu 24,27% dan MAPE untuk data pengujian yaitu 26,23%. Pada data pengujian, sebagai pengukur kehandalan peramalan, nilai MAPE terkecil diperoleh pada bulan Maret 2011 yaitu sebesar 5,29%. Dan MAPE terbesar diperoleh pada bulan Maret 2008 yaitu sebesar 46,10%. Adapun hasil peramalan lengkapnya dapat dilihat pada tabel berikut.

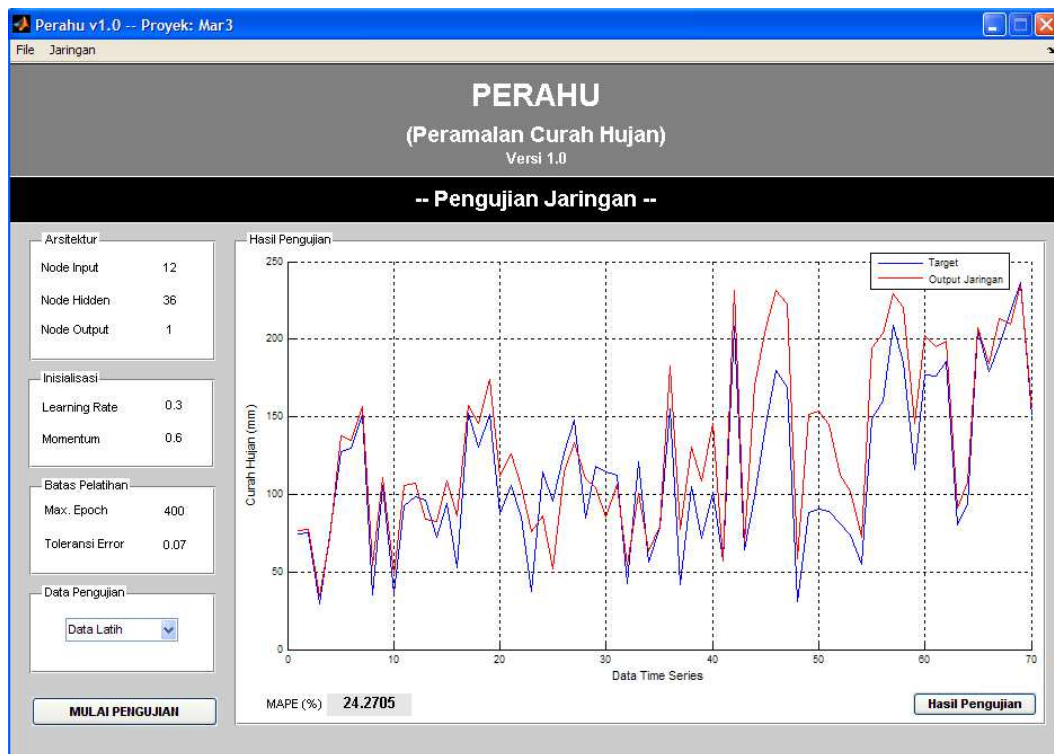
Tabel 4.14 Hasil Peramalan Bulan Maret

No.	Tahun	Nilai Sebenarnya (mm)	Nilai Peramalan (mm)	Error (%)
1.	2004	119,8	155,78	30,03%
2.	2005	115,60	78,29	32,27%
3.	2006	159,9	112,6	29,59%
4.	2007	156,3	103,15	34,00%
5.	2008	197	106,18	46,10%
6.	2009	156,2	188,16	20,46%
7.	2010	115,4	129,33	12,07%
8.	2011	112,2	118,13	5,29%

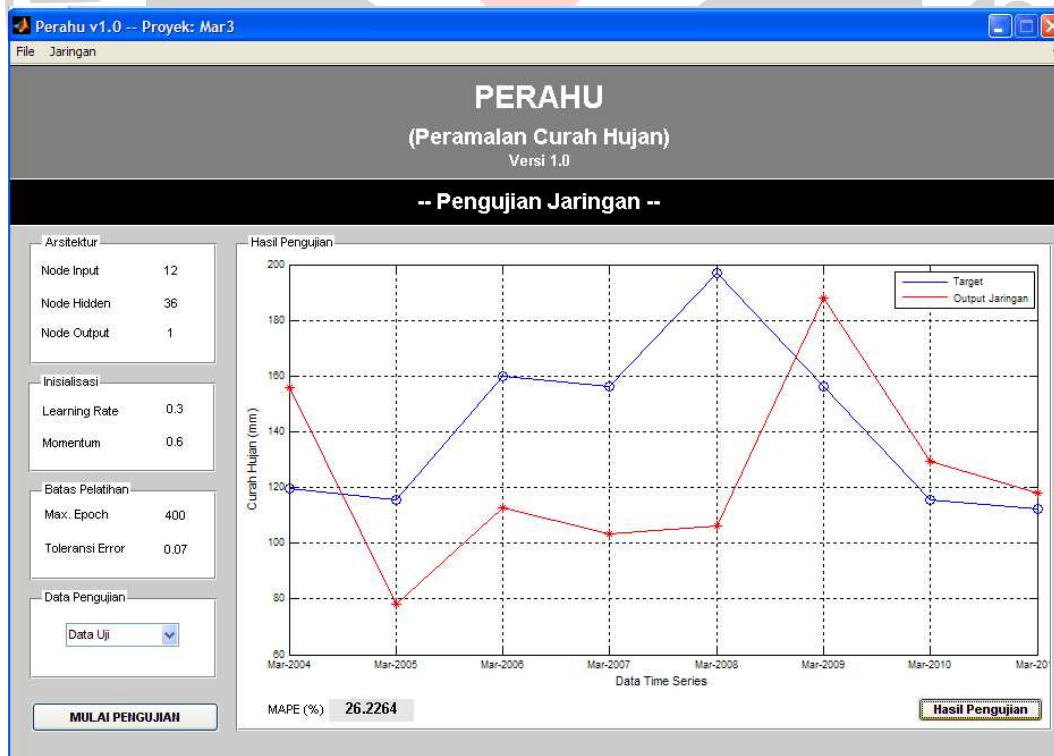
Hasil pelatihan dan pengujian untuk bulan Maret dapat dilihat pada gambar berikut.



Gambar 4.8 Hasil Pelatihan Bulan Maret



Gambar 4.9 Hasil Pengujian dengan Data Pelatihan



Gambar 4.10 Hasil Pengujian dengan Data Pengujian

4.6.3 Analisis Hasil Penelitian

Berdasarkan hasil studi kasus, secara umum jaringan yang dilatih dapat meramalkan nilai curah hujan satu bulan ke depan. Tetapi tingkat akurasinya berbeda-beda. Pada bulan Februari, nilai MAPE peramalan yang dihasilkan tinggi, yaitu 78,69%. Bila ditinjau dari datanya, masih terdapat anomali data dan fluktuasi data yang tidak seimbang. Hal ini dibuktikan dengan besarnya nilai MAPE pada bulan Februari pada studi kasus ke-6, walaupun epoch yang dilakukan sebanyak 1000. Dengan kata lain, jaringan tidak dapat mengenali pola yang diberikan, terlebih lagi meramalkan nilai curah hujan bulan berikutnya.

Selain itu, ada beberapa faktor lain yang mempengaruhi akurasi peramalan, yaitu sebagai berikut.

- a. Pada tahap *data preprocessing* untuk mendeteksi data *outlier*, mungkin masih terdapat data *outlier* yang tidak terdeteksi dengan metode yang digunakan. Karena dengan adanya data *outlier* dapat menjadi penghambat jaringan mengenali pola pelatihan yang diberikan.
- b. Komposisi data pelatihan dengan data pengujian mungkin kurang tepat sehingga sulit untuk mendapatkan bobot yang dapat mengenali pola pelatihan sekaligus dapat meramalkan dengan baik.
- c. Inisialisasi data yang digunakan pada proses pelatihan kurang tepat sehingga bobot optimum tidak tercapai.
- d. Nilai batas toleransi kesalahan pelatihan dan jumlah maksimum epoch sebagai indikator berakhirnya pelatihan kurang tepat sehingga ada kemungkinan terlampauinya bobot optimum ataupun belum tercapai.